

wykład 4

# **Dopasowanie par sekwencji**

dr Jacek Śmietański

[jacek.smietanski@ii.uj.edu.pl](mailto:jacek.smietanski@ii.uj.edu.pl)

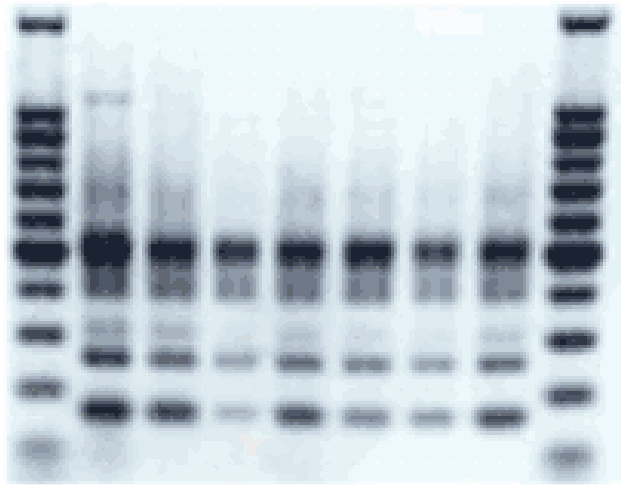
<http://jaceksmietanski.net>

1. Idea i cele dopasowania sekwencji
2. Definicje
3. Macierze kropkowe (dot-plot)
4. Podobieństwo a identyczność
5. Algorytm Needlemana-Wunscha - model liniowy



# Idea i cele dopasowania sekwencji

# Cel porównywania sekwencji



ataaattctttatTTTTgacactcac  
agtcacctggaaaaacccgcttttt  
aaagtacagaaggcttgggtcacaa  
atcactgagaactagagagaaata  
tcgcaaactgtaatagacattaca  
aaaagtttccccagtccttattgt  
cacagtgcattgctacatggcaa

Określenie stopnia podobieństwa między sekwencjami.

Podobieństwo to mierzymy jednak nie miarą stosowaną przy porównywaniu zwykłych danych tekstowych. Interesuje nas podobieństwo ewolucyjne.

Miarą podobieństwa sekwencji (nukleotydowych, aminokwasowych) będzie zatem miara związana z prawdopodobieństwem i ilością mutacji niezbędnych do przeprowadzenia jednej sekwencji w drugą.

Pierwsze pytanie biologa molekularnego, kiedy odkryje nową sekwencję:



*Czy w bazie sekwencji są już sekwencje podobne do mojej?*

- sekwencje są identyczne – nic nowego...
- sekwencja jest podobna (ma „krewnych”) – nowy członek znanej rodziny
- sekwencja ma kilka podobnych regionów, motywów lub domen – można zaproponować funkcję
- nie ma znaczącego podobieństwa – dużo pracy...

## Mutacje:

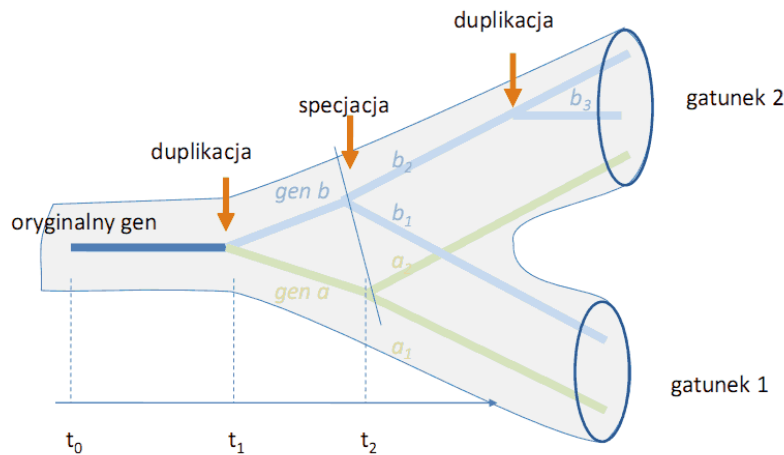
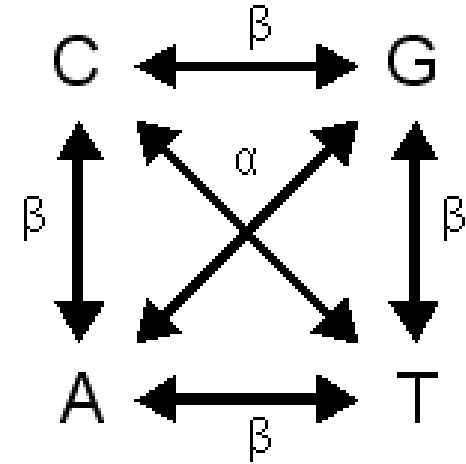
substytucja

$\alpha$  = tranzycja ( $A \leftrightarrow G, C \leftrightarrow T$ )

$\beta$  = transwersja (pozostałe)

insercja

delecja



Białka homologiczne (spokrewnione ewolucyjnie) mają podobne sekwencje.

**Podobieństwo** – miara ilościowa

**Homologia** – określenie jakościowe

wnioskowanie o pokrewieństwie sekwencji na podstawie stopnia ich podobieństwa

**Pytanie:**

Jaki stopień podobieństwa decyduje o homologii?

**I w drugą stronę:**

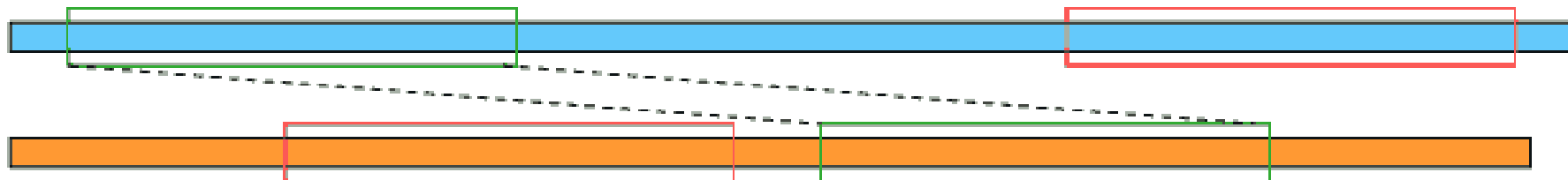
Jakie jest prawdopodobieństwo, że dwie przypadkowe sekwencje będą identyczne?

**dopasowanie globalne** – dopasowanie wzdłuż całej sekwencji  
(np. analiza białek składających się z pojedynczej domeny lub białek homologicznych,  
słabo zróżnicowanych;  
stosujemy dla sekwencji stosunkowo krótkich – np. pojedyncze geny – o zbliżonej  
długości)



**dopasowanie lokalne** – poszukiwanie lokalnych regionów  
o dużym stopniu podobieństwa

(np. analiza białek wielodomenowych, poszukiwanie motywów  
stosujemy dla sekwencji istotnie różniących się długością (np. poszukiwanie krótkich  
motywów w całym genomie); często odległych ewolucyjnie)





# Definicje

Rozważmy sekwencję  $\mathbf{S}$  o długości  $l$  nad alfabetem  $\Sigma$

Oznaczmy:

$\mathbf{S}[i..j]$  - podciąg  $\mathbf{S}$  zaczynający się na pozycji  $i$  i kończący się na pozycji  $j$  (  $\mathbf{S} = \mathbf{S}[0, l-1]$  )

$\mathbf{S}[i]$  - symbol z  $\mathbf{S}$  na pozycji  $i$

$|\mathbf{S}|$  - długość sekwencji  $\mathbf{S}$  (  $|\mathbf{S}| = l$  )

$\epsilon$  - pusta sekwencja (sekwencja o długości 0)

- - symbol przerwy (  $- \notin \Sigma$  )

Dla analizy sekwencji DNA

$$\Sigma = \{A, C, G, T\}$$

Dla analizy sekwencji RNA

$$\Sigma = \{A, C, G, U\}$$

Dla analizy sekwencji aminokwasów

$$\Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$$

W zależności od potrzeb można też używać innych, rozszerzonych alfabetów.

## **DNA**

**A** = adenina

**C** = cytozyna

**G** = guanina

**T** = tymina

**R** = G A (zasada purynowa)

**Y** = T C (zasada pyrimidinowa)

**K** = G T (keto)

**M** = A C (amino)

**S** = G C (silne wiązania)

**W** = A T (słabe wiązania)

**B** = G T C (dowolny, lecz nie A)

**D** = G A T (dowolny, lecz nie C)

**H** = A C T (dowolny, lecz nie G)

**V** = G C A (dowolny, lecz nie T)

**N** = A G C T (dowolny)

## **RNA**

**A** = adenina

**C** = cytozyna

**G** = guanina

**U** = uracyl

**R** = G A (zasada purynowa)

**Y** = U C (zasada pyrimidinowa)

**K** = G U (keto)

**M** = A C (amino)

**S** = G C (silne wiązania)

**W** = A U (słabe wiązania)

**B** = G U C (dowolny, lecz nie A)

**D** = G A U (dowolny, lecz nie C)

**H** = A C U (dowolny, lecz nie G)

**V** = G C A (dowolny, lecz nie U)

**N** = A G C U (dowolny)

# Rozszeszony alfabet dla białek

A	<b>Ala</b> alanina	P	<b>Pro</b> prolina
B	<b>Asx</b> kw. asparaginowy/asparagina	Q	<b>Gln</b> glutamina
C	<b>Cys</b> cysteina	R	<b>Arg</b> arginina
D	<b>Asp</b> kw. asparaginowy	S	<b>Ser</b> seryna
E	<b>Glu</b> kw. glutaminowy	T	<b>Thr</b> treonina
F	<b>Phe</b> fenyloanina	U	selenocysteina
G	<b>Gly</b> glicyna	V	<b>Val</b> walina
H	<b>His</b> histydyna	W	<b>Trp</b> tryptofan
I	<b>Ile</b> izoleucyna	Y	<b>Tyr</b> tyrozyna
K	<b>Lys</b> lizyna	Z	<b>Glx</b> kw.glutaminowy/glutamina
L	<b>Leu</b> leucyna	X	<b>Xxx</b> dowolny
M	<b>Met</b> metionina		
N	<b>Asn</b> asparagina		

## Definicja:

Niech  $S_0, S_1$  – sekwencje nad alfabetem  $\Sigma$  o długościach odpowiednio  $l_0$  i  $l_1$ . **Globalnym dopasowaniem pary sekwencji** (*global pairwise sequence alignment*) nazywamy macierz  $M$  o wymiarach  $2 \times n$ , gdzie  $n \geq \max\{l_0, l_1\}$  taką, że

$\forall 0 \leq k \leq 1, 0 \leq i \leq n-1$ :

$M[k][i] = -$  lub  $M[k][i] = S_k[p], p \in \{0, \dots, l_k - 1\}$

$M[k][i] = - \Rightarrow M[1-k][i] \neq -$

$\forall i < j \ (M[k][i] = S_k[p] \text{ i } M[k][j] = S_k[q] \Rightarrow p < q)$

$\forall 0 \leq p \leq l_k - 1 \ \exists j \in \{0, \dots, n-1\} : M[k][j] = S_k[p]$

## Przykład:

$S_0 = \text{TAGACTAG}$

$S_1 = \text{ACGTATG}$

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

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

**S1 = ATTG**

**S2 = ATGG**

Podaj przykłady dopasowania.

Zwykle jesteśmy w stanie zaproponować wiele różnych dopasowań.

Które dopasowanie jest optymalne? Czy możemy to określić jednoznacznie?

Będziemy szukali dopasowania najlepszego, tj. takiego, które minimalizuje skojarzoną z tym dopasowaniem funkcję kary.

# Macierze kropkowe



# Idea metody dot plot: tworzenie macierzy

§ Take 2 sequences and write each along one side of a 2D matrix

§ Every place where the sequences match, place a dot

§ To obtain an alignment, find long diagonal runs

§ An example:

seq #1: **ATTGCCCATG**

seq #2: **ATGGCCATTG**

	A	T	T	G	C	C	C	A	T	G
A	*							*		
T		*	*						*	
G				*						*
G				*						*
C					*	*	*			
C					*	*	*			
A	*							*		
T		*	*						*	
T		*	*						*	
G				*						*

Miejsca zgodności nukleotydów (aminokwasów)  
zaznaczamy kropkami.

# Idea metody dot plot: tworzenie macierzy (2)

§ Take 2 sequences and write each along one side of a 2D matrix

§ Every place where the sequences match, place a dot

§ To obtain an alignment, find long diagonal runs

§ An example:

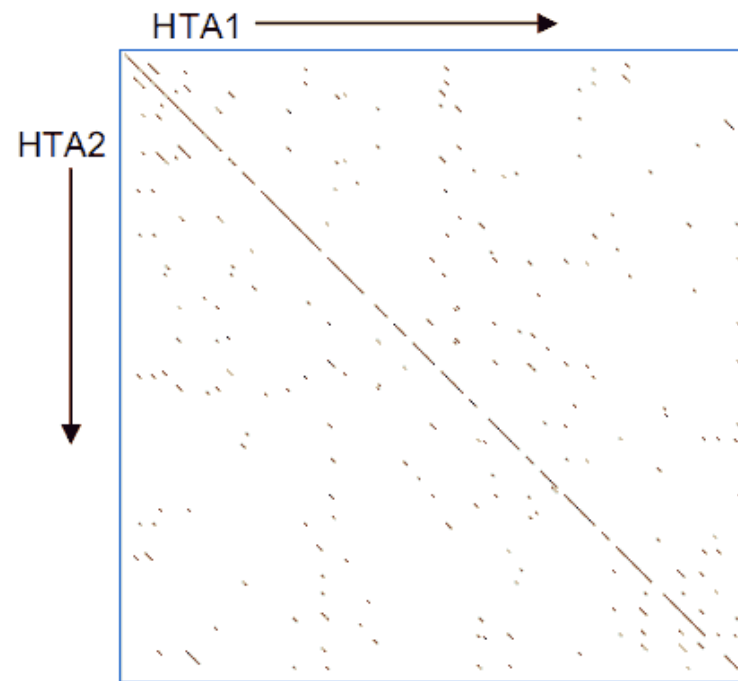
seq #1: **ATTGCCCATG**

seq #2: **ATGGCCATTG**

	A	T	T	G	C	C	C	A	T	G
A	*							*		
T		*	*						*	
G				*						*
G				*						*
C					*	*	*			
C					*	*	*			
A	*							*		
T		*	*						*	
T		*	*						*	
G				*						*

Dopasowanie – tam, gdzie kropki tworzą linię ciągłą na przekątnej.

# Przykład: porównanie dwóch spokrewnionych genów

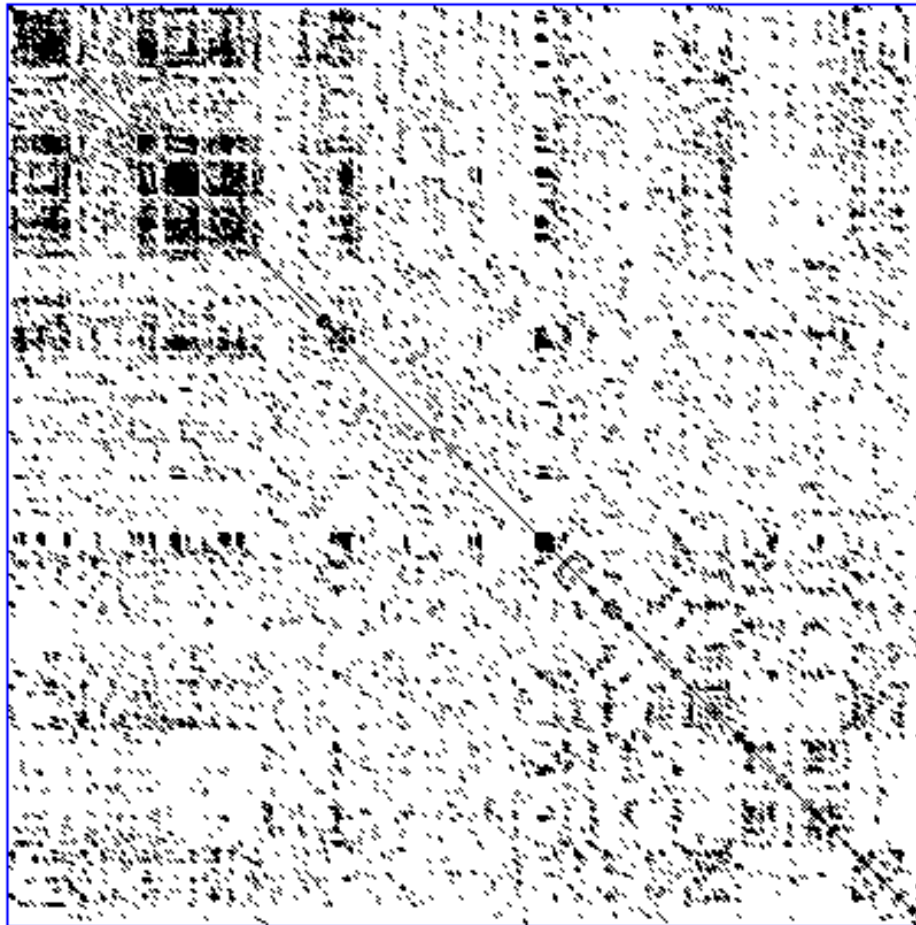


§ Dot plot comparison of yeast HTA1 and HTA2 genes (both code for histone H2A)

§ Window size = 5 (each dot represents a window of 5 nucleotides that match)

Image generated using: <http://www.vivo.colostate.edu/molkit/dnadot/>

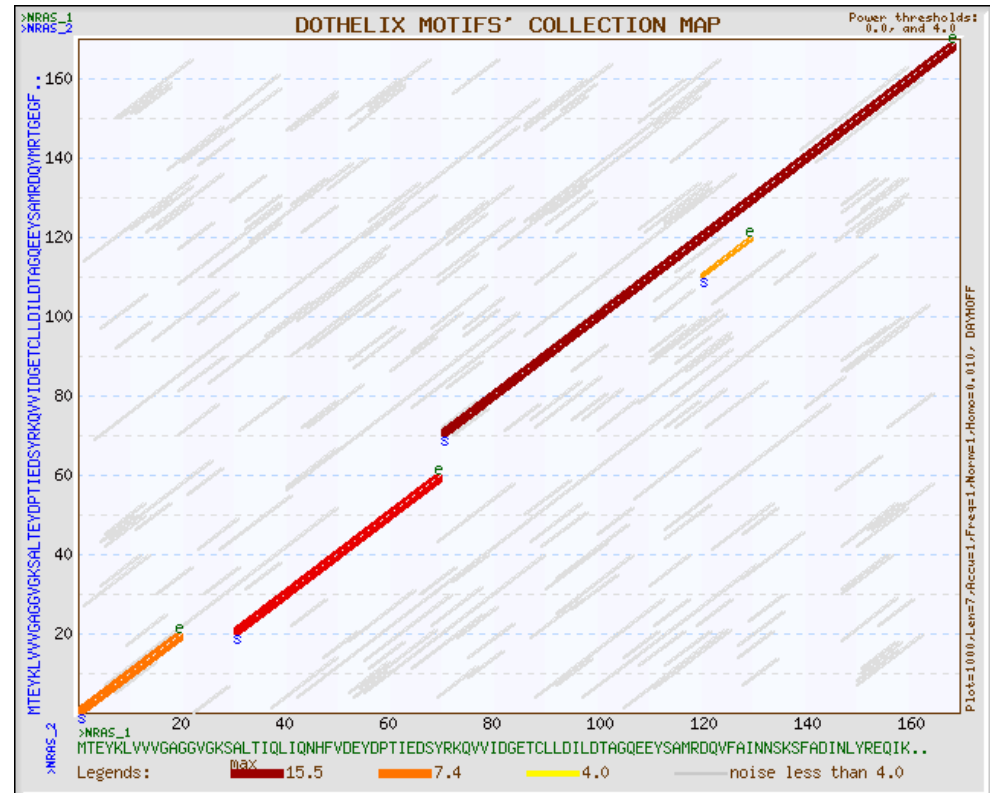
Jak będzie wyglądała macierz kropkowa dla pary sekwencji z dużą liczbą powtarzających się fragmentów?



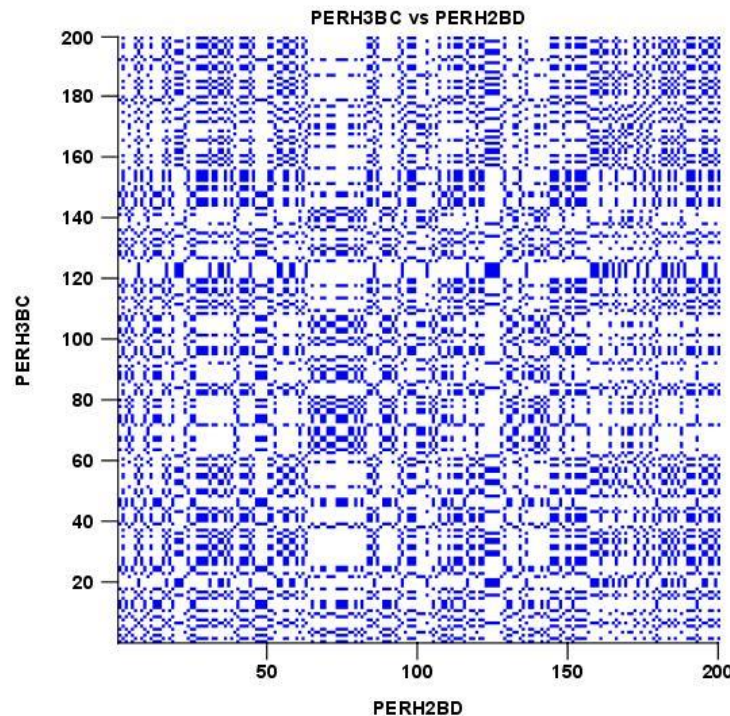
- Szum

(pewnym rozwiązaniem problemu może tu być metoda okna)

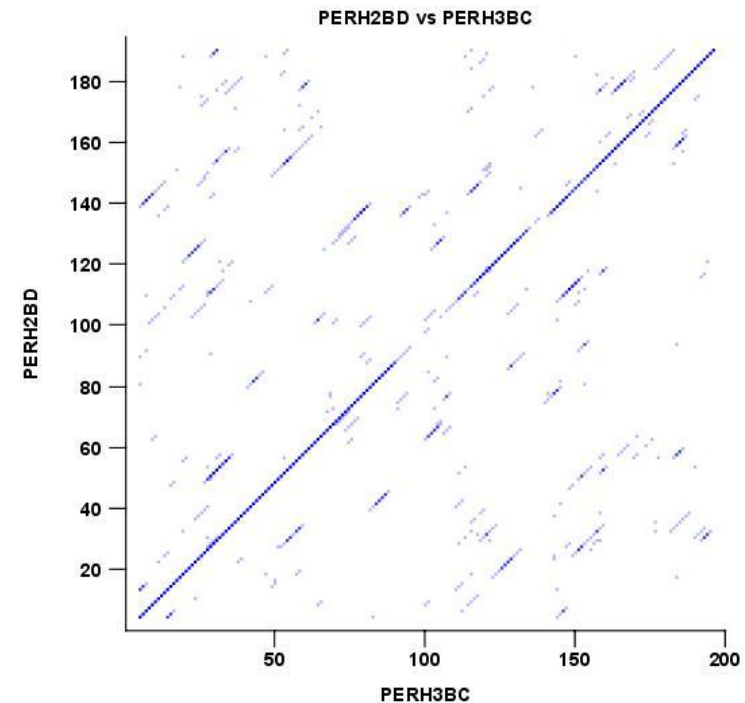
- Brak możliwości jakościowej oceny dopasowania



Idea: zaznaczanie punktu tylko jeżeli identyczność występuje dla całego podciągu sekwencji o określonej długości (parametr nazywany wielkością okna)



rozmiar okna: 1



rozmiar okna: 9

# Metoda okna – przykład (1)


Sequence 2	C								
	C								
	T								
	A								
	A								
	A								
	G								
	G								
		G	G	A	A	A	T	C	C
Sequence 1									

Jak wypełnimy macierz?

- a) okno wielkości 1
- b) okno wielkości 3

a) rozmiar okna: 1

Sequence 2	C								
	C								
	T								
	A								
	A								
	A								
	G								
	G								
		G	G	A	A	A	T	C	C
		Sequence 1							

 zgodność

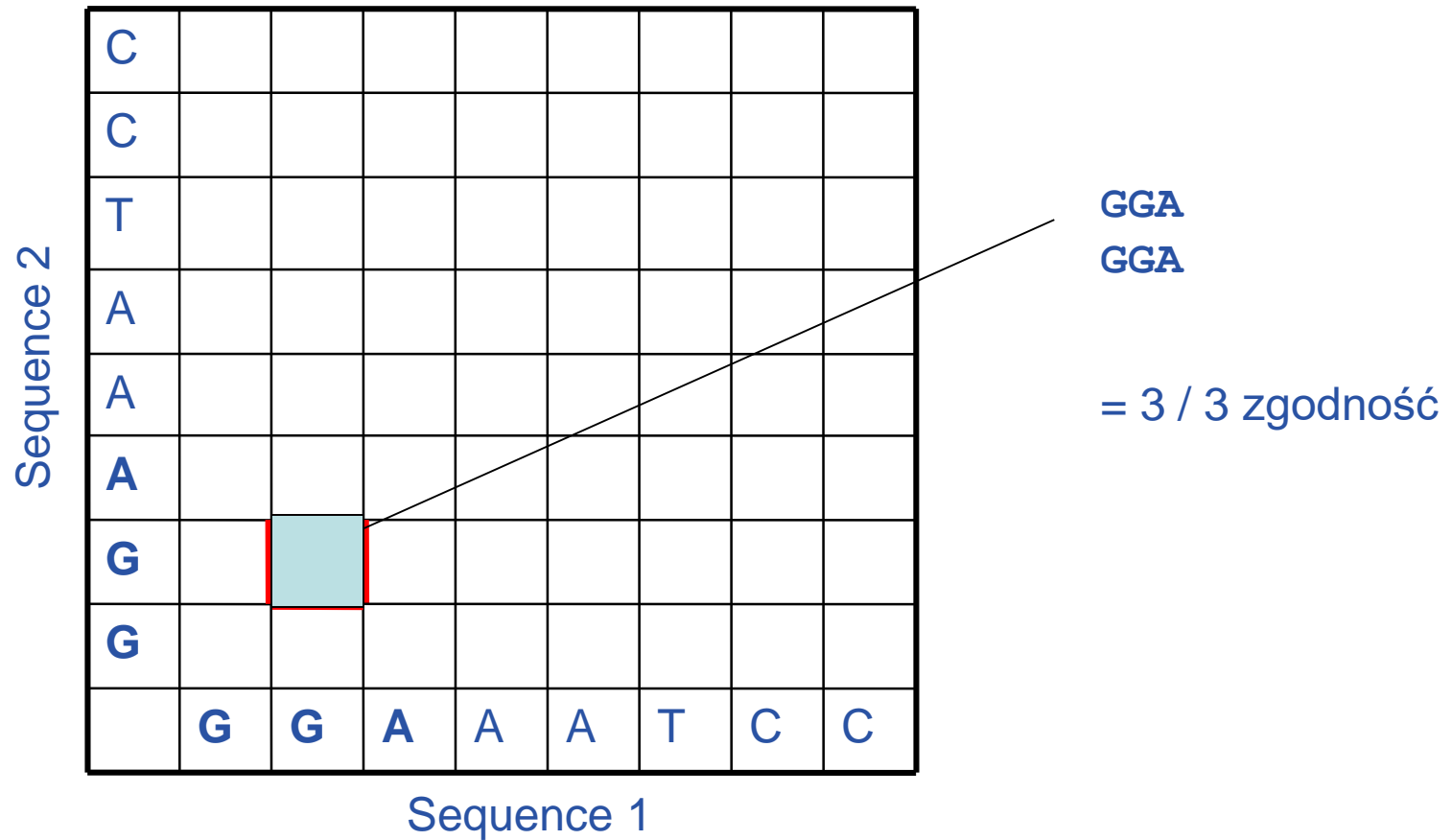


## b) rozmiar okna: 3

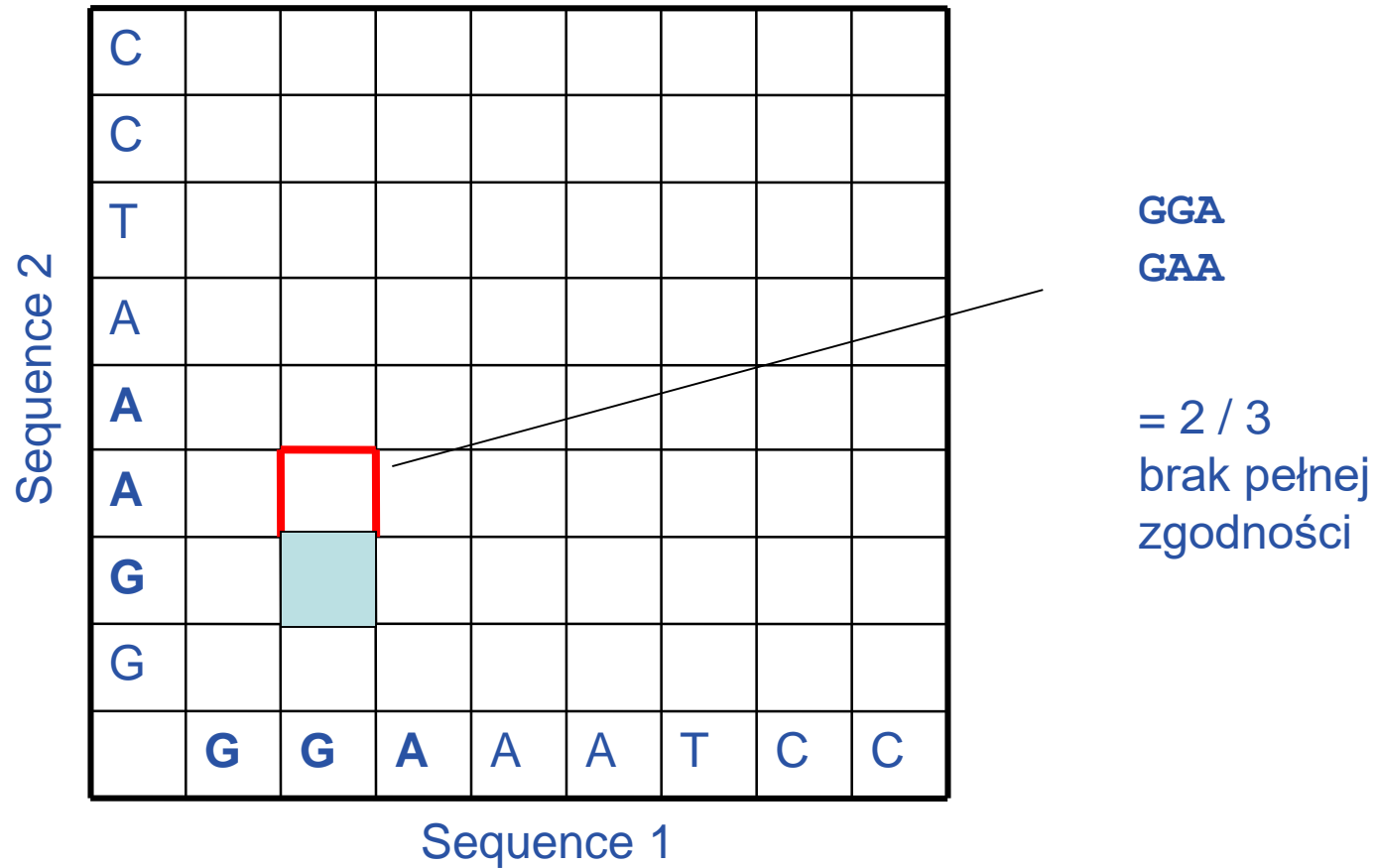
Sequence 2	C								
	C								
	T								
	A								
	A								
	A								
	G								
	G								
	G	G	A	A	A	T	C	C	
		Sequence 1							

obszary brzegowe  
pomijamy

b) rozmiar okna: 3



b) rozmiar okna: 3



b) rozmiar okna: 3 - wyjściowa macierz

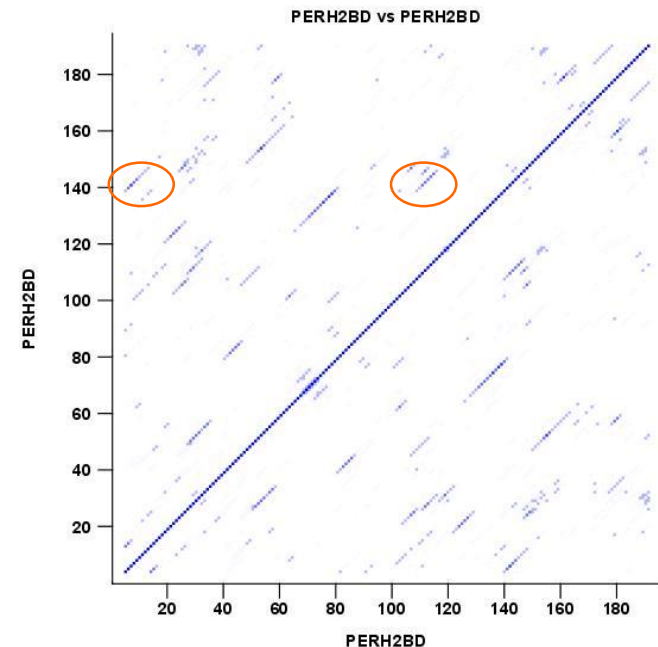
Sequence 2

C								
C							*	
T						*		
A					*			
A				*				
A			*					
G		*						
G								
	G	G	A	A	A	T	C	C

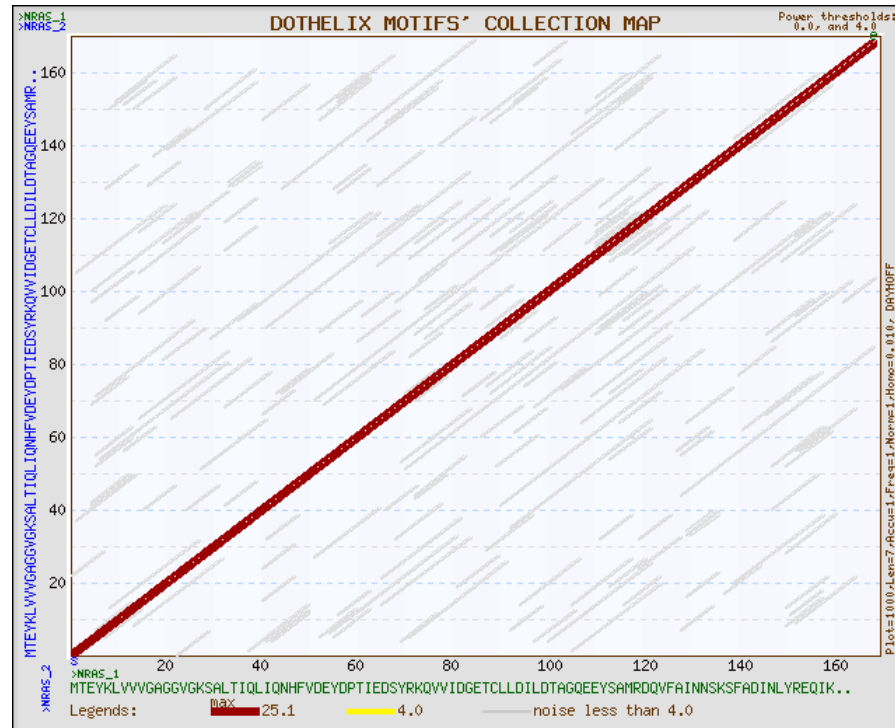
Sequence 1

## Identyfikacja obszarów:

- identyczności między dwiema sekwencjami
- insercje-delecje: introny
- motywy (porównanie sekwencji samej ze sobą)
- odwrócone powtórzenia

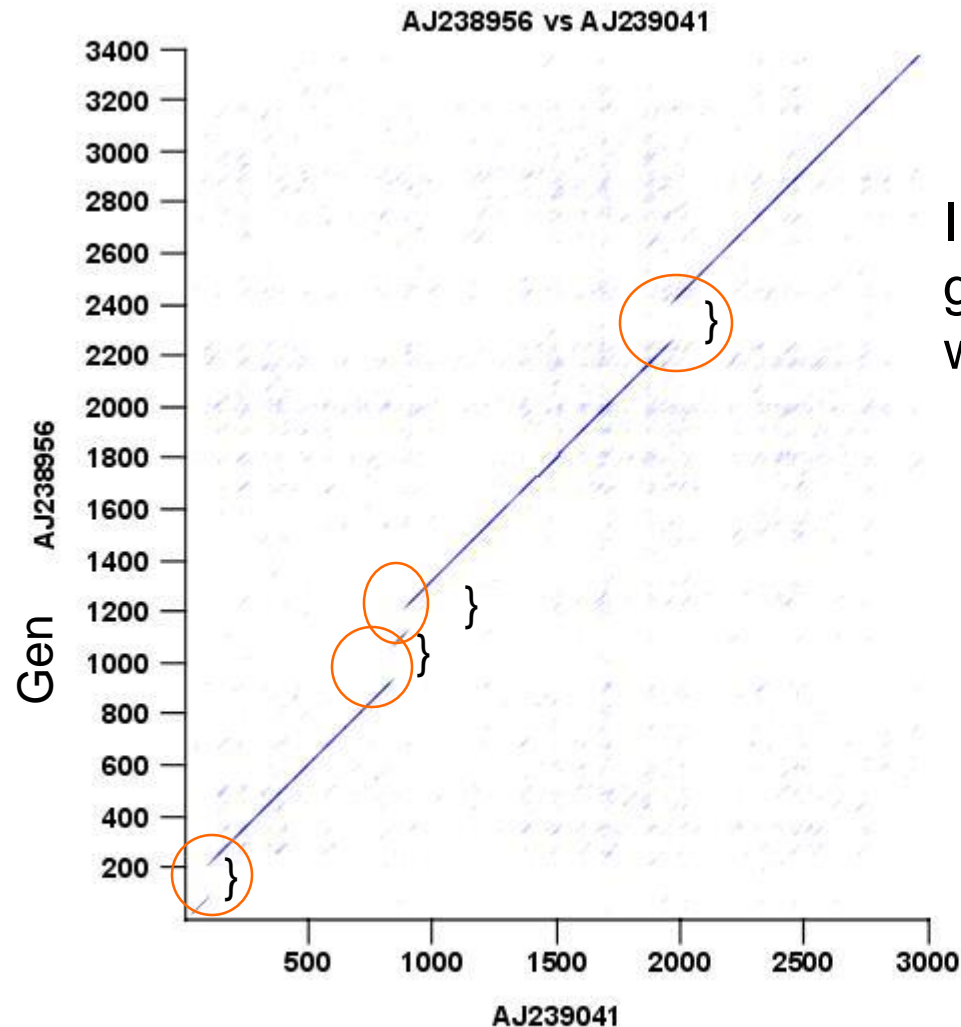


- a) identyfikacja motywów (powtarzających się krótkich fragmentów sekwencji)



- b) identyfikacja komplementarnych odwróconych powtórzeń (*inverted repeats*) - wykorzystywana do określenia struktury drugorzędowej RNA (np. spinka do włosów)

# Przykład zastosowania – identyfikacja intronów



Introny (niekodujące fragmenty genów) są usuwane z mRNA w procesie jego dojrzewania

dojrzałe mRNA

## Dotmatcher

<http://emboss.bioinformatics.nl/cgi-bin/emboss/dotmatcher>

## Dottup

<http://emboss.bioinformatics.nl/cgi-bin/emboss/dottup>

## Dothelix

<http://www.genebee.msu.su/services/dhm/advanced.html>

## Gepard

<http://cube.univie.ac.at/gepard>

## MatrixPlot

<http://www.cbs.dtu.dk/services/MatrixPlot/>



# Podobieństwo a identyczność

dla sekwencji nukleotydowych – pojęcia synonimiczne

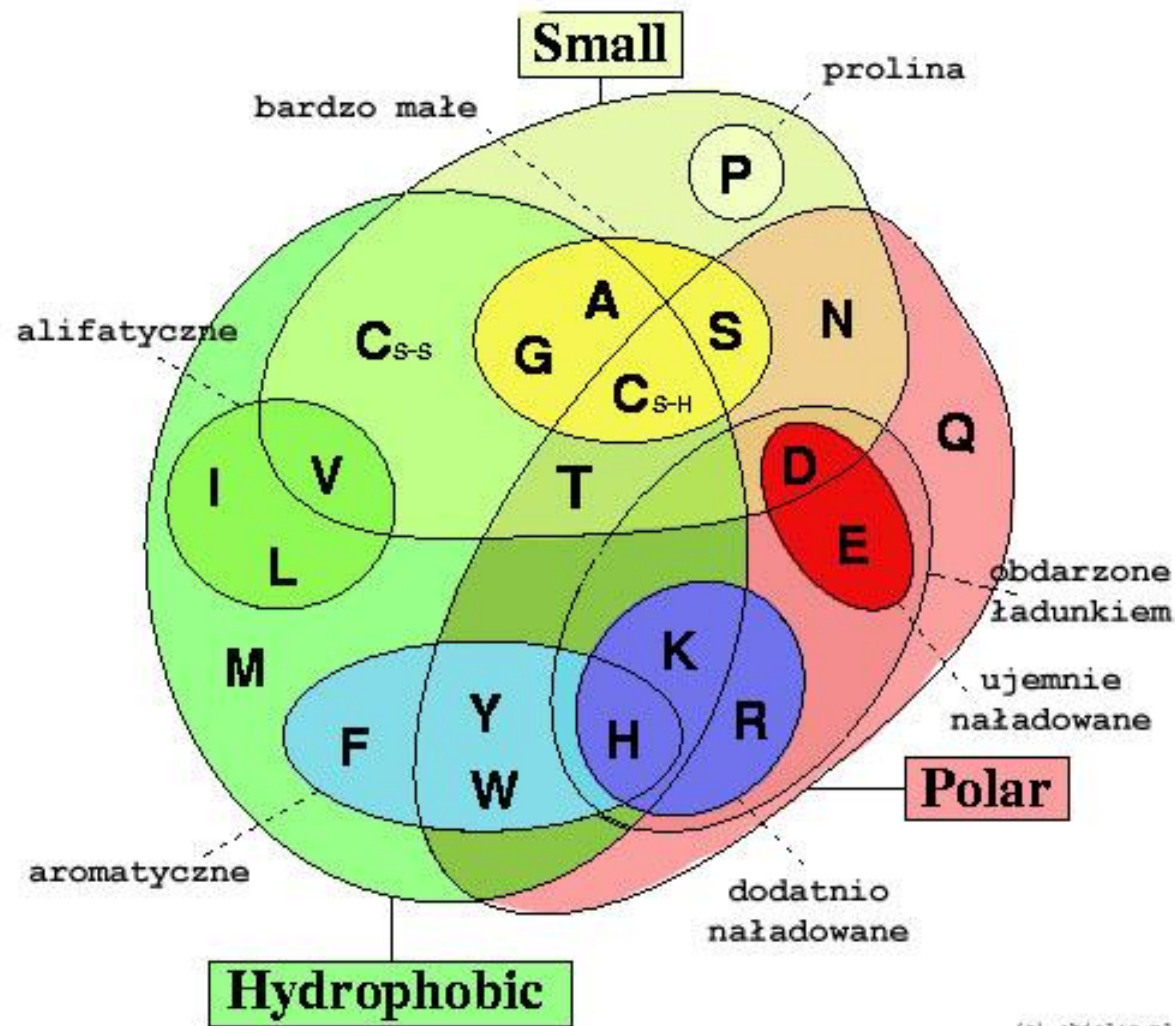
dla sekwencji białkowych:

**identyczność:** procent identycznych reszt aminokwasowych skojarzonych ze sobą

**podobieństwo:** procent przyrównanych reszt, które wykazują zbliżone właściwości fizykochemiczne

# Właściwości aminokwasów

Klasyfikacja  
aminokwasów  
białkowych  
ze względu na ich  
właściwości  
fizykochemiczne



S – podobieństwo

I - identyczność

*metoda 1:*

$$S\% = \frac{2L_s}{L_a + L_b} 100\%$$

$$I\% = \frac{2L_i}{L_a + L_b} 100\%$$

*metoda 2:*

$$S\% = \frac{L_s}{L_a} 100\%$$

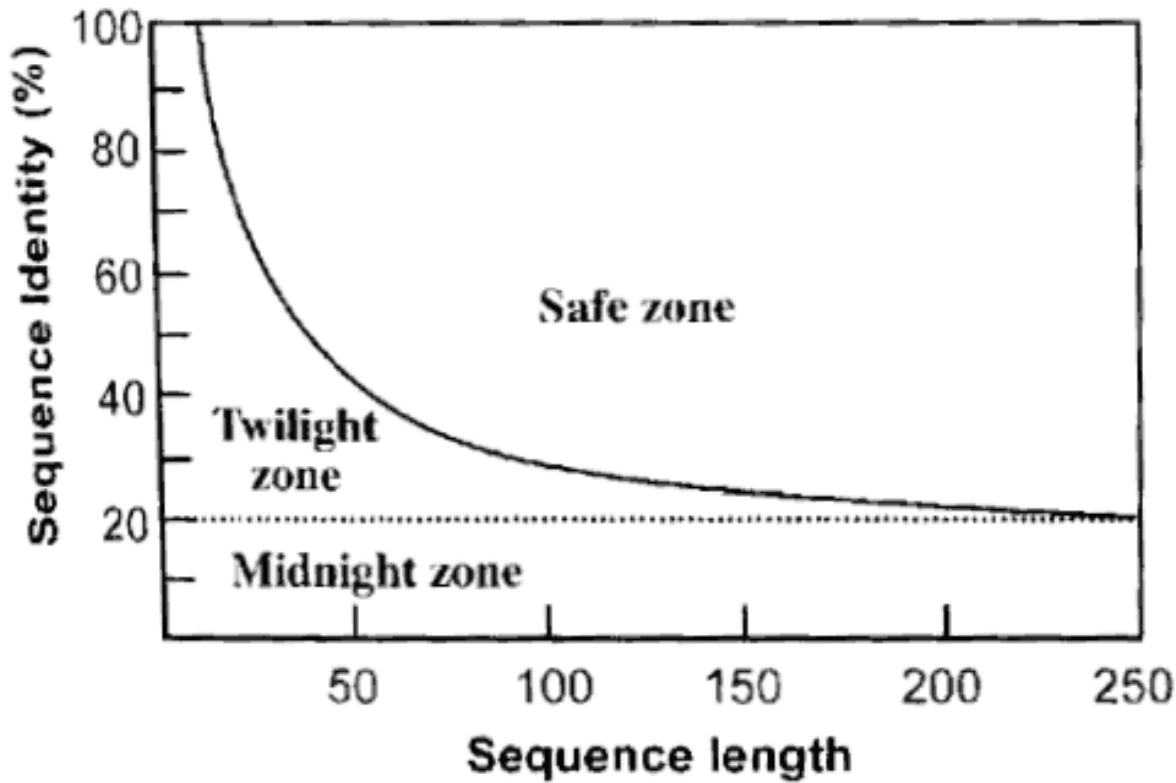
$$I\% = \frac{L_i}{L_a} 100\%$$

$L_s$  – liczba przyrównanych reszt podobnych

$L_i$  – liczba przyrównanych reszt identycznych

$L_a$   $L_b$  – długości sekwencji  $a$  i  $b$ ,  $L_a \leq L_b$

# Strefy przyrównań sekwencji białkowych



- strefa bezpieczna (możemy wnioskować o homologii)
- strefa cienia (homologia możliwa, ale wymaga potwierdzenia)
- strefa ciemności (wnioskujemy o braku homologii)

# Algorytm Needlemana-Wunscha **model liniowy**

Technika konstrukcji algorytmów mająca zastosowanie do problemów o własności optymalnej podstruktury: optymalne rozwiązanie problemu jest funkcją optymalnych rozwiązań podproblemów.

Idea algorytmu polega zatem na podziale zadanego problemu na mniejsze podproblemy.

W algorytmie wyczerpującym (sprawdzenie wszystkich możliwych kombinacji) rozwiązanie całego zagadnienia wymaga zwykle wielokrotnego rozwiązania tych samych podproblemów; zysk obliczeniowy w programowaniu dynamicznym otrzymujemy dzięki temu, że **każdy podproblem rozwiązujemy tylko raz**, zapamiętując te częściowe rozwiązania w pomocniczej strukturze danych (zwykle w macierzy).

Wykorzystujemy macierz pomocniczą: podobnie jak w metodzie dot plot zapisujemy dwie porównywane sekwencje – jedną w kolumnie, drugą w wierszu.

Rozmiar macierzy:  $(i+1)*(j+1)$ , gdzie  $i, j$ : odpowiednio długości sekwencji  $S_1$  i  $S_2$

Macierz wypełniamy liczbami (punktacja) stosując metodę programowania dynamicznego, tj. oceniając wartość optymalnego dopasowania dla krótszych podsekwencji.

W każdym kolejnym kroku badamy jak się zmieni punktacja, jeżeli do wcześniej dopasowanej sekwencji dodamy nukleotyd lub przerwę.

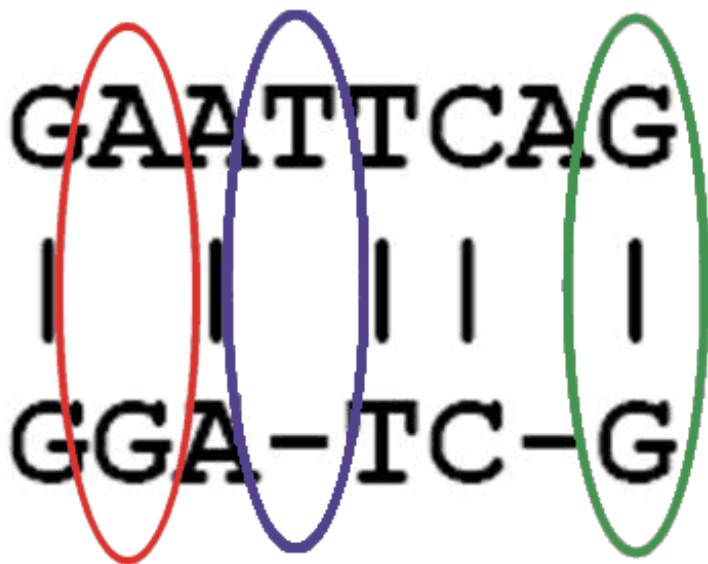


Wykorzystujemy system punktacji, pamiętając, że przy ocenie dopasowania musimy uwzględnić nie tylko pozycje, gdzie nukleotydy (bądź aminokwasy) pokrywają się ze sobą (zgodność), ale też niezgodności w dopasowaniu (efekt substytucji), jak i występowanie przerw (efekt insercji lub delecji).

Im wyższa punktacja, tym lepsze dopasowanie.

Punkty liczymy dla każdej kolumny dopasowania. Są trzy możliwości:

- **zgodność** (w danej kolumnie są te same nukleotydy / aminokwasy)
- **niezgodność** (w danej kolumnie są różne nukleotydy / aminokwasy)
- **przerwa** (w danej kolumnie występuje symbol „-”)



Punktacja za zgodność/niezgodność będzie nazywana ogólnie „punktacją za dopasowanie”, a punktację przerw nazywa się zwykle „karą za przerwę”.

Ostateczna wartość dopasowania pary sekwencji jest sumą punktów obliczonych dla wszystkich kolumn dopasowania.

## 1. dopasowanie:

- a) zgodność (wartość dodatnia, np. +1)  
niezgodność (zwykle wartość ujemna, np. -1)
- b) wartości punktowe bierzemy z macierzy substytucji

## 2. przerwa (wartość ujemna, np. -2)

- a) jednakowa wartość za każdą przerwę (model liniowy)
- b) dodatkowa kara za rozpoczęcie przerwy (model afiniczny)

1a) Przy dopasowywaniu sekwencji **nukleotydowych** stosuje się ustalone punkty za zgodność i niezgodność niezależnie od tego, które nukleotydy są do siebie dopasowane.

1b) Przy dopasowywaniu sekwencji **aminokwasowych**, punktacja za dopasowanie zwykle uzależniona jest od rodzaju dopasowanych aminokwasów – wartości punktowe bierze się z tzw. „macierzy substytucji”.

Sposób punktowania jak i konkretne wartości punktacji nie są sztywno narzucone. Ustala się je w zależności od potrzeb jako parametry algorytmu.

**liniowy model kar za przerwy** – każda przerwa traktowana jest tak samo;

**afiniczny model kar za przerwy** – preferuje łączenie pojedynczych przerw w większe zbitki.

Czasem na końcach sekwencji dopuszczamy brak kary za przerwę.

Do przemyślenia:

Jaki wpływ na dopasowanie mają konkretne wartości poszczególnych elementów punktacji?

Co się stanie gdy kara będzie za duża?

Za mała?

*Wartości brzegowe:*

$$H_{0,0} = 0$$

$$H_{i,0} = i * g$$

$$H_{0,j} = j * g$$

*Wartości macierzy dla  $1 \leq i \leq |S_1|$  i  $1 \leq j \leq |S_2|$ :*

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j) \\ H_{i-1,j} + g \\ H_{i,j-1} + g \end{cases}$$

H – macierz punktacji;

s – funkcja dopasowania symboli  $a_i$  i  $b_j$

(punkty dodatnie za zgodność, ujemne za niezgodność

lub wg macierzy substytucji [omówione zostaną one na kolejnym wykładzie])

g – kara za przerwę (wartość ujemna)

$S_1 = \text{TGCTCGTA}$

$S_2 = \text{TTCATA}$

Punktacja:

zgodność: +5

niezgodność: -2

przerwa: -6

1. Wypełniamy zerowy wiersz i kolumnę.

	T	G	C	T	C	G	T	A	
T	0	-6	-12	-18	-24	-30	-36	-42	-48
T	-6								
T	-12								
C	-18								
A	-24								
T	-30								
A	-36								

# Działanie algorytmu – przykład (2)

$S_1 = \text{TGCTCGTA}$

$S_2 = \text{TTCATA}$

Punktacja:

zgodność: +5

niezgodność: -2

przerwa: -6

2. Dla każdej komórki wstawiamy największą spośród trzech wartości:

a) wartość komórki wyżej po przekątnej + punktacja za dopasowanie

b) wartość komórki po lewej + kara za przerwę

c) wartość komórki powyżej + kara za przerwę

- a)  $0 + 5 = 5$
- b)  $-6 - 6 = -12$
- c)  $-6 - 6 = -12$

Zapamiętujemy, skąd się  
wzięła wpisana wartość  
(na diagramie czerwona  
strzałka)

	T	G	C	T	C	G	T	A	
T	0	-6	-12	-18	-24	-30	-36	-42	-48
T	-6	5							
C	-12								
A	-18								
T	-24								
A	-30								
A	-36								

$S_1 = \text{TGCTCGTA}$

$S_2 = \text{TTCATA}$

Punktacja:

zgodność: +5

niezgodność: -2

przerwa: -6

3. Wartość w prawym dolnym rogu macierzy jest wartością optymalnego dopasowania.

Wyznaczamy je na podstawie ścieżki od tego punktu do początku macierzy.

	T	G	C	T	C	G	T	A	
T	0	-6	-12	-18	-24	-30	-36	-42	-48
T	-6	5	-1	-7	-13	-19	-25	-31	-37
T	-12	-1	3	-3	-2	-8	-14	-20	-26
C	-18	-7	-3	8	2	3	-3	-9	-15
A	-24	-13	-9	2	6	0	1	-5	-4
T	-30	-19	-15	-4	7	4	-2	6	0
A	-36	-25	-20	-10	1	5	2	0	11



$S_1 = \text{TGCTCGTA}$

$S_2 = \text{TTCATA}$

Punktacja:

zgodność: +5

niezgodność: -2

przerwa: -6

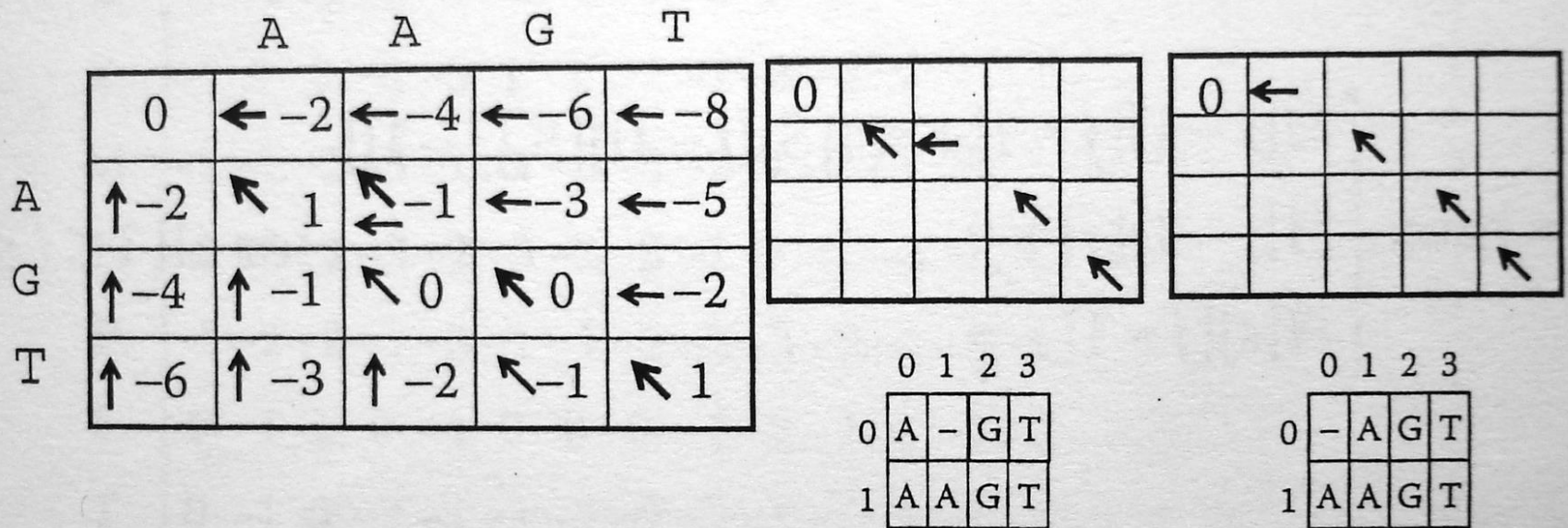
## 4. Optymalne dopasowanie:

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

		T	G	C	T	C	G	T	A
	0	-6	-12	-18	-24	-30	-36	-42	-48
T	-6	5	-1	-7	-13	-19	-25	-31	-37
T	-12	-1	3	-3	-2	-8	-14	-20	-26
C	-18	-7	-3	8	2	3	-3	-9	-15
A	-24	-13	-9	2	6	0	1	-5	-4
T	-30	-19	-15	-4	7	4	-2	6	0
A	-36	-25	-20	-10	1	5	2	0	11

Pionowe kreski w zapisie pomiędzy górną a dolną sekwencją nie są konieczne, zwiększają jednak czytelność, wskazując obszary identyczności w dopasowaniu.

Odtwarzamy kierunki, dzięki którym uzyskaliśmy dane do macierzy. Każda ścieżka od  $H[|S_1|][|S_2|]$  do  $H[0][0]$  jest najlepszym dopasowaniem (może istnieć więcej niż jedno).



Punktacja w powyższym przykładzie: zgodność +1, niezgodność -1, przerwa -2. Istnieją tu dwa optymalne dopasowania.

## Praktyczna implementacja:

		A	A	G	T		
	0	←-2	←-4	←-6	←-8	0	
A	↑-2	↖1	↖-1	←-3	←-5		
G	↑-4	↑-1	↖0	↖0	←-2		
T	↑-6	↑-3	↑-2	↖-1	↖1		

	0	1	2	3
0	A	-	G	T
1	A	A	G	T

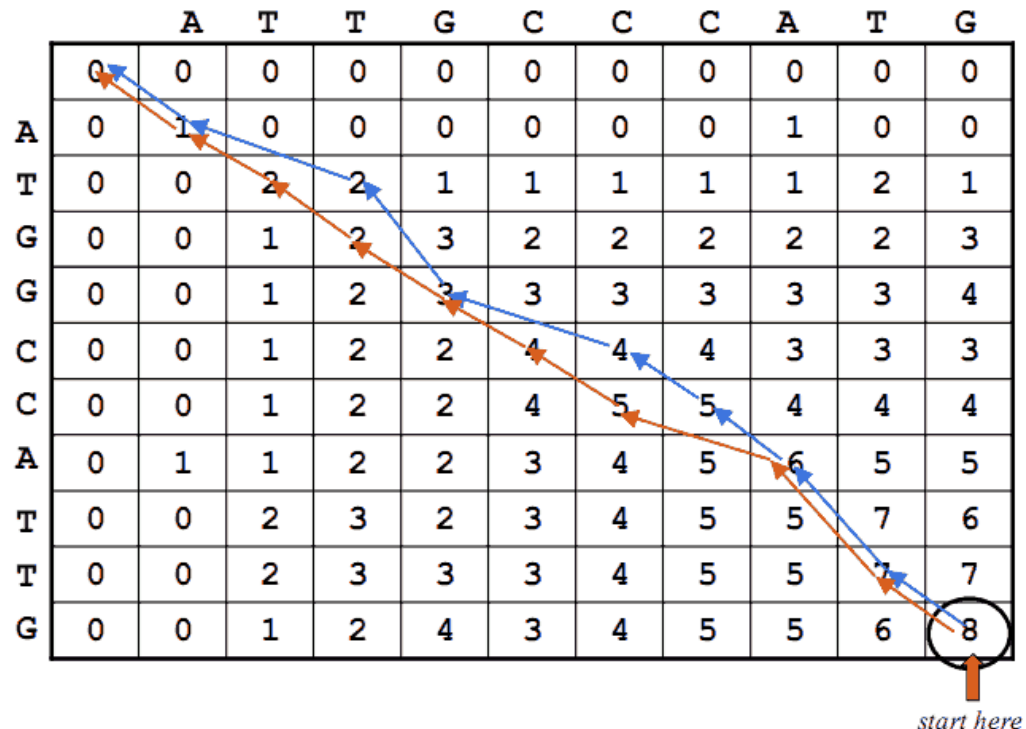
  

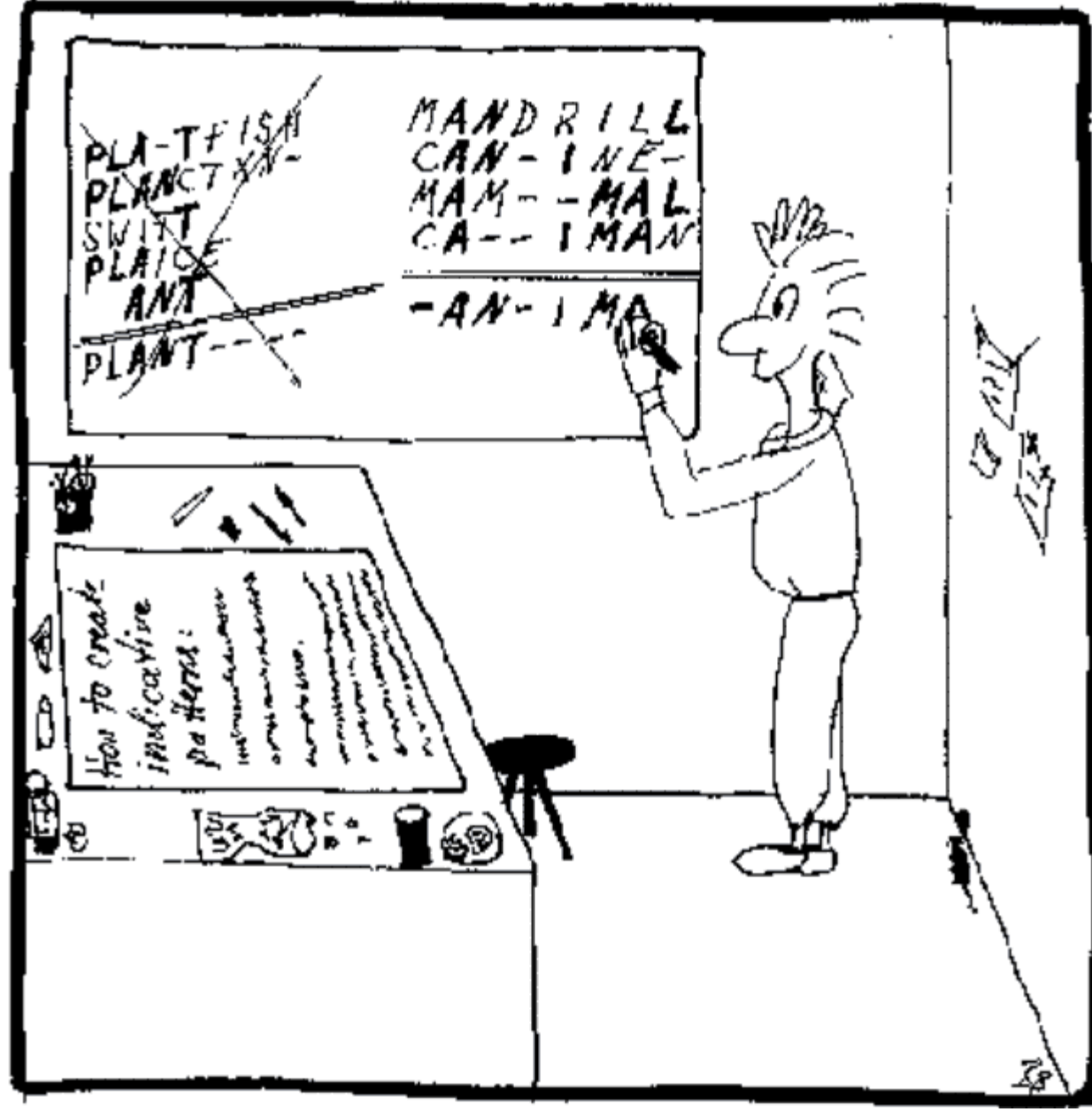
	0	1	2	3
0	-	A	G	T
1	A	A	G	T

- a) W trakcie konstrukcji macierzy H, zapamiętujemy w dodatkowych polach (macierz pomocnicza) kierunki, z których przyszliśmy.
- Rozwiązanie polecane przy niezbyt dużych macierzach, gdy nie ogranicza nas dostępna pamięć.
- b) Obliczamy te kierunki (proste równania) na bieżąco podczas rekonstrukcji.

Jak efektywnie zaimplementować rekonstrukcję wszystkich możliwych optymalnych ścieżek?

W implementacji tej wersji algorytmu, w zerowym wierszu i kolumnie wstawiamy same zera, a ścieżkę rekonstruujemy od największej wartości w ostatnim wierszu lub kolumnie (niekoniecznie musi to być prawy dolny róg).





Brigitte Boeckmann / 1995