MSA

Porównywanie wielu sekwencji

- **MSA – jeden z kluczowych problemów bioinformatyki (biologii obliczeniowej)**
- **problem**
  - **trudno wyznaczyć kryterium porównywania**
- **pomysły**
  - **ilość identycznych pozycji w sekwencjach o identycznej długości**
  - **suma dopasowań PSA dla wszystkich par**

MSA – uliniowianie wielu sekwencji

- **Przykład sekwencji aminokwasów:**
  - **HBA_HUMAN** ***(prefiks ludzkiej hemoglobiny)***

VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTT
KTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPN
ALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEF
TPAVHASLDKFLASVSTVLTSKYR

- **Baza sekwencji aminokwasów *BaliBase* 2.0 i *BaliBase* 3.0**

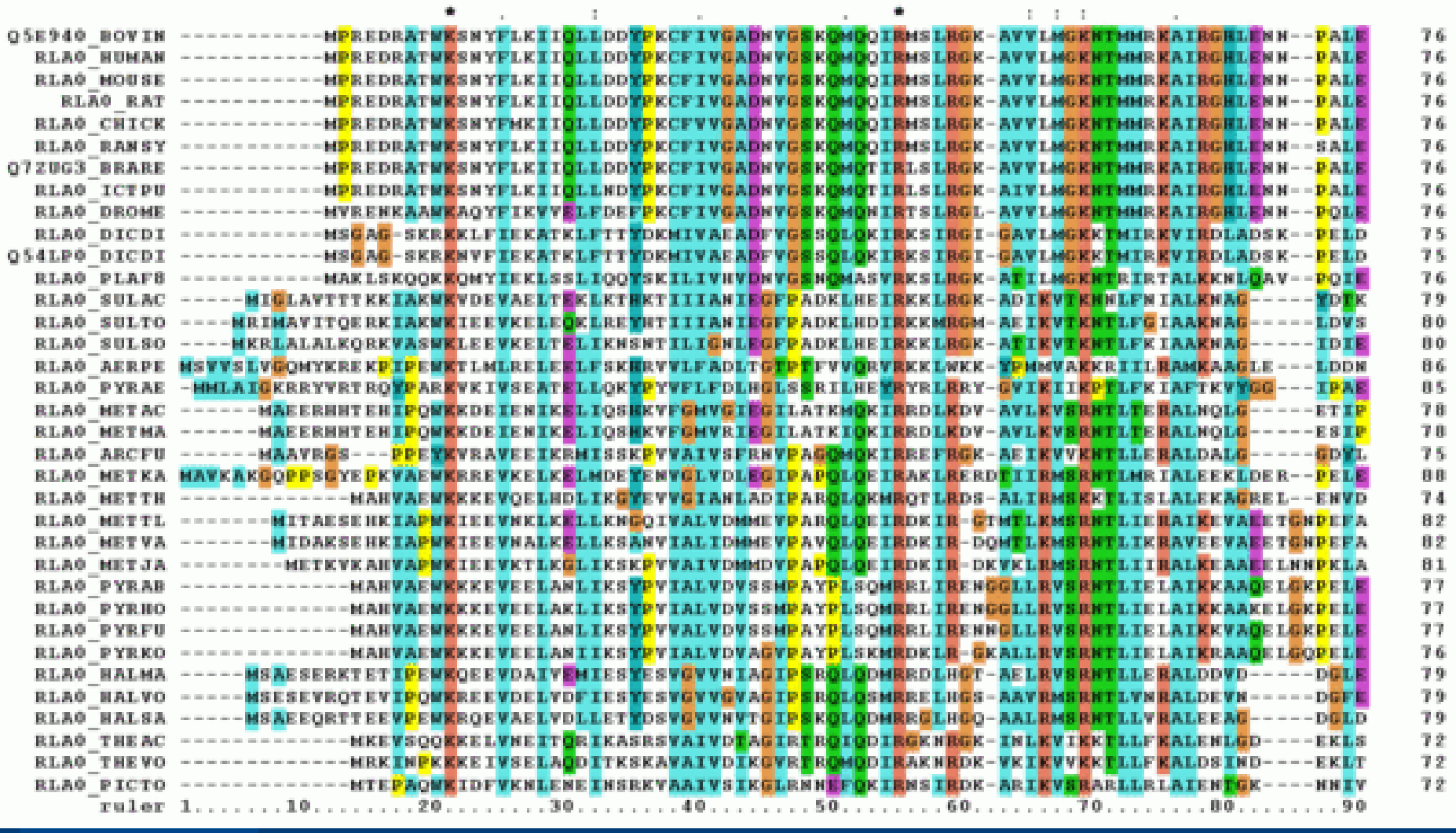
MSA – uliniowianie wielu sekwencji

- **DNA, RNA lub sekwencje aminokwasów**
- **Zasady uliniowiania (analogiczne do przypadku dwóch sekwencji):**
  - **Każda z uliniowionych sekwencji ‘wraca” do postaci oryginalnej po usunięciu odstępów.**
  - **Sekwencje po uliniowieniu mają taką samą długość.**
  - **Żadna z kolumn nie jest zbudowana wyłącznie z odstępów.**

Przykład uliniowienia

- **Program CLUSTAL W**

```
LGB2_LUPLU   VPQ--NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSK-GVADAHFPV
MYG_PHYCA    EAEMKASEDLKKHGVTVLTALGAILKKKG--HHEAELKPLAQS---HATKHKIPIKYLEF
GLB5_PETMA   ADQLKKSADVRWHAERIINAVNDAVASMD--DTEKMSMKLRDLSGKHAKSFQVDPQYFKV
HBB_HUMAN    PDAVMGNPKVKAHGKKVLGAFSDGLAHLD--NLKGTFATLSEL---HCDKLHVDPENFRL
HBB_HORSE    PGAVMGNPKVKAHGKKVLHSFGEGVHHLD--NLKGTFAALSEL---HCDKLHVDPENFRL
HBA_HUMAN    -----GSAQVKGHGKKVADALTNAVAHVD--DMPNALSALSDL---HAHKLRVDPVNFKL
HBA_HORSE    -----GSAQVKAHGKKVGDALTLAVGHLD--DLPGALSNLSDL---HAHKLRVDPVNFKL
                       .  .: *.  :   .                    *  :   *  .  :    : .
```



From Wikipedia. Generated with ClustalX

Metoda ewolucyjno-progresywna

Metoda ewolucyjno-progresywna

- **metoda 2-etapowa**
- **etap 1. - ewolucyjny**
  - **dopasowywanie kolumn całkowicie identycznych**
  - **znajdowanie optymalnego tzw. „wstępnego uliniowienia”**
  - **etap wykonywany rekurencyjnie**
- **etap 2. - progresywny**
  - **uliniowienie obszarów między kolumnami zidentyfikowanymi w etapie 1.**

Etap ewolucyjny

dopasowywanie kolumn całkowicie identycznych, przykład:

```
MAAFCP
MACFMCP
MACMFCP
```

wszystkie możliwe kolumny zgodne

1	1	1	1	2	3	4	5	5	5	5	6
1	1	5	5	2	2	4	3	3	6	6	7
1	4	1	4	2	2	5	3	6	3	6	7

Etap ewolucyjny

1	1	1	1	2	3	4	5	5	5	5	6
1	1	5	5	2	2	4	3	3	6	6	7
1	4	1	4	2	2	5	3	6	3	6	7

blok kolumn

- kolumny tworzą blok jeśli we wszystkich wierszach różnica w indeksach wynosi jeden (większy indeks – mniejszy indeks)
- blok może mieć dowolną długość
  - w szczególności pojedynczą kolumną również można traktować jako blok

1	2	5	6
1	2	6	7
1	2	6	7

Etap ewolucyjny

1	1	1	1	2	3	4	5	5	5	5	6
1	1	5	5	2	2	4	3	3	6	6	7
1	4	1	4	2	2	5	3	6	3	6	7

wstępne uliniowienie

- szereg bloków spełniający następujące warunki
  - dowolny indeks może wystąpić w wierszu tylko raz
  - w każdym wierszu indeksy są w porządku rosnącym
- powyższe warunki gwarantują, że na podstawie wstępnego uliniowienia można zbudować pełne uliniowienie (zachowując ustalone kolumny identyczne)

1	2	5	6	
1	2	3	7	
1	2	6	7	

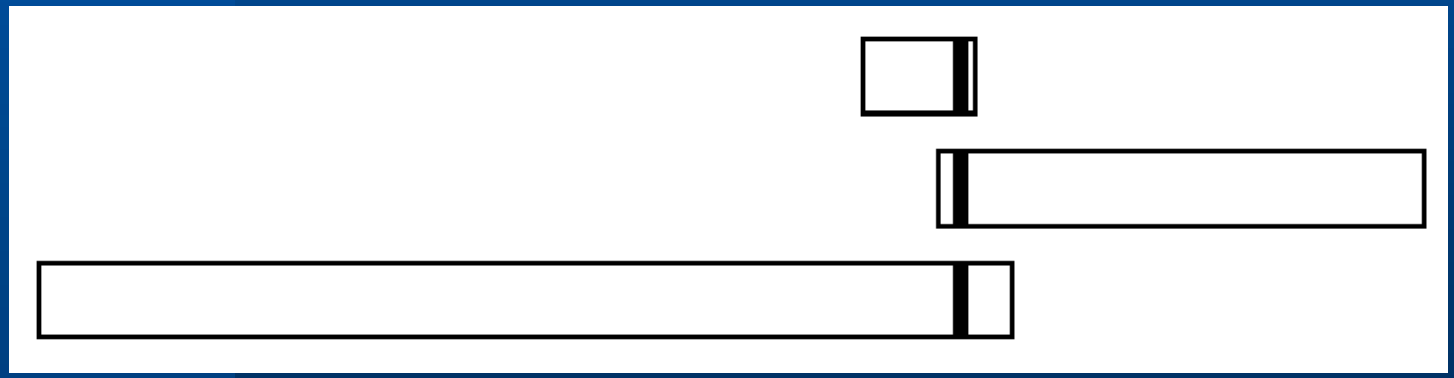
1	2	4	5	6
1	2	4	6	7
1	2	5	6	7

Etap ewolucyjny

1	1	1	1	2	3	4	5	5	5	5	6
1	1	5	5	2	2	4	3	3	6	6	7
1	4	1	4	2	2	5	3	6	3	6	7

Kolumny/bloki szkodliwe

- intuicyjnie możemy określić taką kolumnę jako łączącą „zbyt” odległe części różnych sekwencji
- kolumna taka, uniemożliwia bardzo często lepsze dopasowanie innych kolumn identycznych



Etap ewolucyjny

- **bliskie optymalnemu uliniowienie z wymuszeniem uzgodnienia kolumny symboli T**

```
MARAFCPMWAAAFCP-----T--MAAFCP------
----------------MAARFTCPMAAFCPMAAFCP
MRAAFCPMW-AAFCPMAA-FTCP-----------
```

- **uliniowienie tych samych sekwencji bez uzgadniania symboli T**

```
MARAF-CPMWAAAFCPTMAAF-CP
MAARFTCPM--AAFCP-MAAF-CP
MRAAF-CPMW-AAFCP-MAAFTCP
```

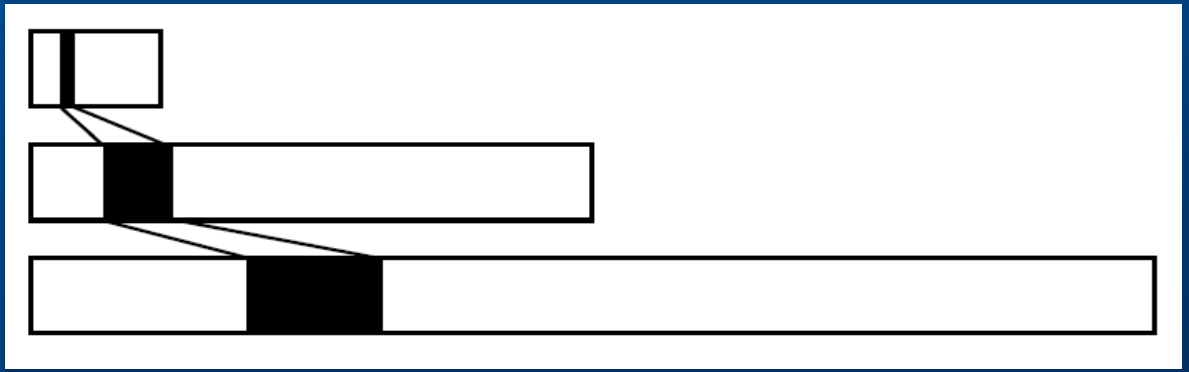
Etap ewolucyjny

- **zadania algorytmu ewolucyjnego**
  - **znalezienie optymalnego wstępnego uliniowienia (NIE PEŁNEGO uliniowienia) ➔ mniejsze wymagania czasowe i pamięciowe**

- **budowa populacji startowej**
  - **czas budowy musi być „racjonalny”**
  - **wprowadzenie do populacji startowej reprezentatywnego podzbioru możliwych kolumn identycznych**
    - **użycie wszystkich (z wszystkich części sekwencji) symboli z sekwencji**
  - **unikanie szkodliwych kolumn**
    - **ew. późniejsza ich eliminacja**

Budowa populacji startowej

- **metodę charakteryzują dwa podstawowe parametry**
  - **c_{max} – górny limit (w przybliżeniu) liczby zidentyfikowanych kolumn identycznych**
  - **$w_{\%}$ – szerokość tzw. „okna przeszukiwania”**
    - **symbole tworzące kolumnę identyczną nie mogą pochodzić z dowolnych części sekwencji**
    - **każdy symbol pochodzi z aktywnego okna przeszukiwania danej sekwencji**



Budowa populacji startowej

- **względna długość okna przeszukiwania (w stosunku do dł. sekwencji) jest taka sama dla wszystkich sekwencji**
- **analogicznie względna pozycja środka okna (względem początku sekwencji)**

- **z każdego okna, losowo, wybierany jest jeden symbol**
- **jeśli wszystkie symbole są identyczne, tworzona jest kolumna identyczna**
  - **nie jest sprawdzana unikalność kolumny**
- **czynność jest wykonywana $\left\lceil \frac{c_{\max}}{m} \right\rceil$ razy dla każdego symbolu (okna szerokości jednego symbolu) wyróżnionej sekwencji**
  - **gdzie m – dł. wyróżnionej sekwencji (np. najkrótszej)**

Budowa populacji startowej

- zbieranie informacji (tworzenie wstępnych uliniowień)

```
dla_każdego (a w A) {
        dla_każdego (p w P) {
                jeżeli  (a można dołączyć na koniec p) {
                        dołącz a do p;
                        jeżeli można
                                złącz a z ostatnim blokiem w p;
                        przejdź do następnego a;
                }
        }
        stwórz nowe p z a;
        dołącz p do P;
}
posortuj P zgodnie z wartością funkcji przystosowania;
wybierz co najwyżej cp najlepszych osobników;
```

A – znaleziony zbiór kolumn identycznych (w porządku znajdowania)

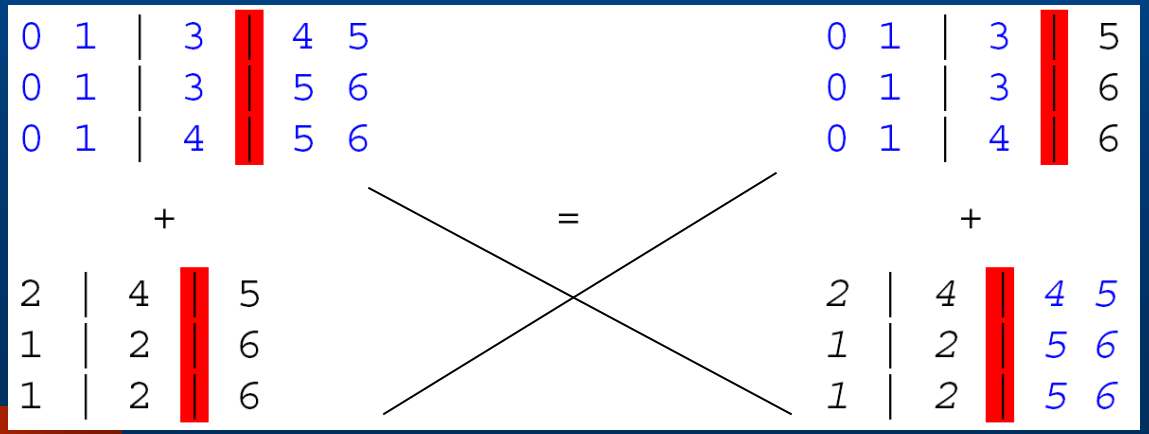
P – populacja startowa, początkowo pusta

c_p – nominalny rozmiar populacji startowej

Algorytm ewolucyjny

- **populacja startowa** (c_{max}=4000, $w_{\%}$=0.04)
  - $c_p = (m_a * n) / 10$,
    m_a – śr. dł. sekwencji, n – ilość sekwencji
  - $c_p >= 100$ oraz $c_p <= 400$

- **tylko jeden operator genetyczny - krzyżowanie**



Algorytm ewolucyjny

- **krzyżowanie**
  - **jednopunktowe**
  - **losowe punkty cięcia (możliwe przed pierwszym i za ostatnim blokiem)**
  - **punkt cięcia nigdy nie rozdziela bloku**
  - **po wymianie informacji sprawdzana jest możliwość złączenia bloków sąsiadujących z punktem cięcia**
  - **„lepszy” z potomków musi być lepszy od obojga rodziców**
  - **jeżeli potomek nie reprezentuje poprawnego wstępnego uliniowienia to jest odrzucany**
  - **domyślne prawdopodobieństwo krzyżowania = 0.4**

Algorytm ewolucyjny

- **funkcja przystosowania**

$$fitness(p) = 100 \times \frac{col(p)}{(len_{min}(p))^{\alpha}}$$

col(p) **– ilość kolumn identycznych w osobniku *p***
len*$_{min}$*(p) **– minimalna długość uliniowienia powstałego na podstawie uliniowienia wstępnego reprezentowanego przez osobnika *p***
α **– wykładnik określający istotność karania na powstawanie nadmiernie długich uliniowień (=20)**

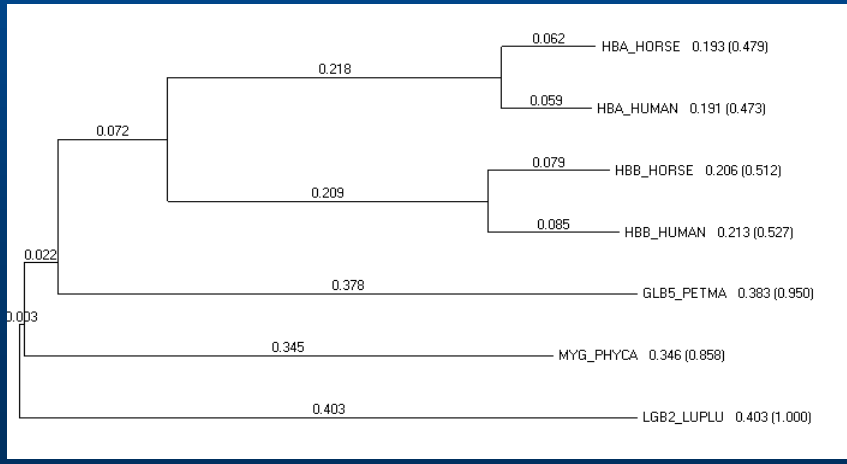
Algorytm ewolucyjny

- warunki stopu
  - przystosowanie najlepszego osobnika nie zmieniło się od 40 generacji
  - osiągnięto limit 1000 generacji
- wywołania rekurencyjne dla obszarów między blokami (w najlepszym z osobników)
- koniec rekurencji
  - alg. ewolucyjny nie znalazł żadnej kolumny identycznej
  - minimalna odległość między danymi blokami jest <= 20

Algorytm progresywny

- **uruchamiany dla obszarów między blokami zidentyfikowanymi przez alg. ewolucyjny**

- **implementacja zbliżona do ClustalW**
  - **PSA**
  - **drzewo filogenetyczne budowane metodą neighbor-joining (z ukorzenianiem metoda mid-point rooting)**



Testy

- **Na podstawie referencyjnych baz BAliBASE**

	BAliBASE 2.01	BAliBASE 3.0
publication date	2000	2005
number of test cases	141	218
number of sequences in test case	3 - 28	4 - 142
length of sequences	49 - 993	49 - 7923

- **bazy udostępniają zarówno testowe zestawy sekwencji, jak i gotowe uliniowienia tych zestawów**

Ocena uliniowienia

- **miara SPS (Sum-of-Pair Score)**

N - ilość sekwencji

$$\sum_{1 \le i < j \le N} sim(S_i^{\#}, S_j^{\#})$$

- **SPS reprezentuje koszt uliniowienia – im mniejszy tym lepiej**

- **miara CS (Column Score)**
  - **ilość kolumn identycznych w stosunku do dł. uliniowienia**
- **CS reprezentuje stopień zgodności kolumn - im wyższy tym lepiej**

Ocena uliniowienia

- **Zbiór_1: GOP=10, GEP=0.2, BLOSUM62**
- **Zbiór_2: GOP=10, GEP=0.2, PAM250**
- **Wszystkie wyniki podawane są jako średni stosunek miar w odniesieniu do rezultatów dla uliniowień z bazy referencyjnej.**
- **PC z 1,7 GHz, 1GB RAM, ograniczenia czasowe i pamięciowe pojedynczego testu: (1 godzina, 1GB).**

Wyniki (ver 2.01)

RESULTS OBTAINED FOR BALIBASE VER. 2.01 TEST CASES.

	the average SPS ratio	the average CS ratio	sum of the execution times	% of successfully completed test cases	the average length of alignment ratio
ClustalW 1.83	101.2	89.7	90	100.0	96.6
MUSCLE 3.6	99.8	95.2	65	100.0	99.1
MAFFT 5.8	99.7	99.7	24	100.0	100.4
DIALIGN 2.2.1	94.4	77.0	289	100.0	113.0
T-Coffee 4.45	99.2	95.0	1732	100.0	101.3
SAGA 0.95	101.9	77.1	51503	90.8	94.6
E-P ($w_{\%} = 0.01$)	104.1	81.5	47	100.0	100.1
E-P ($w_{\%} = 0.02$)	104.7	86.0	43	100.0	101.2
E-P ($w_{\%} = 0.04$)	105.5	92.7	38	100.0	102.0

- **Wszystkie poza SAGA, 100% skuteczne**
- **Wszystkie metody progresywne są na zbliżonym poziomie.**
- **DIALIGN najlepszy w SPS ale kosztem słabego CS, długich uliniowień i długiego czasu.**
- **E-P nieco szybsza niż metody progresywne CLUSTAL W i MUSCLE, znacznie szybsza niż SAGA (genetyczna).**
- **E-P porównywalna (choć nieco słabsza) do metod progresywnych.**

Wyniki (ver. 3.0)

RESULTS OBTAINED FOR BALIBASE VER. 3.0 TEST CASES.

	the average SPS ratio	the average CS ratio	sum of the execution times	the average length of alignment ratio
ClustalW 1.83	103.6	64.9	2902	94.3
MUSCLE 3.6	101.1	84.1	3276	98.9
MAFFT 5.8	100.5	83.0	350	102.8
DIALIGN 2.2.1	91.8	58.9	15689	129.8
E-P ($w_{\%} = 0.04$)	104.6	74.2	1492	97.3
E-P ($w_{\%} = 0.08$)	104.0	95.0	902	98.9
E-P ($w_{\%} = 0.12$)	102.6	105.6	825	101.2
E-P ($w_{\%} = 0.16$)	101.8	119.8	795	103.1
E-P ($w_{\%} = 0.20$)	100.5	123.1	768	104.8
E-P ($w_{\%} = 0.24$)	99.6	126.5	757	107.7
E-P ($w_{\%} = 0.28$)	98.7	134.2	777	108.7
E-P ($w_{\%} = 0.32$)	97.4	139.3	792	110.6
E-P ($w_{\%} = 0.50$)	94.6	138.5	913	115.6

- **SAGA – zbyt duże wymagania czasowe**
- **E-P → jakość porównywalna do metod progresywnych.**
- **Względem miary SPS dla $w_{\%}$ >= 0,12 E-P jest skuteczniejsza niż CLUSTAL W, a dla w% >= 0,2 niż MUSCLE i MAFFT.**
- **Względem miary CS, dla prawie wszystkich $w_{\%}$ E-P jest skuteczniejsza niż pozostałę metody.**
- **Uliniowienia E-P są dłuższe.**

Podsumowanie

- **Tradycyjne podejście genetyczne, w którym osobniki reprezentują pełne uliniowienia jest zbyt kosztowne.**
- **Przedstawiona metoda E-P łaczy w sobie elastyczność AG oraz szybkość i dokładność metod progresywnych.**
- **W efekcie, stanowi ona alternatywę do czysto genetycznych oraz czysto progresywnych metod.**
- **Poprzez właściwy dobór $w_{\%}$ możliwe jest ustanowienie równowagi pomiedzy SPS, CS oraz długością uliniowienia.**

Pytania?