

Założmy, że mamy dopasować takie 4 sekwencje:

AGTCGTAG		w1	A	G	T	C	G	T	A	G
ATCGTCCG	→ dopasowujemy więc, np. tak: →	w2	A	-	T	C	G	T	C	G
GTAG		w3	-	-	-	-	G	T	A	G
GTAGAG		w4	-	G	T	A	G	-	A	G

w1-w4 : wiersze od 1 do 4 (przypada się dalej).

Teraz trzeba by jakoś ocenić jakość tego dopasowania. Potrzebujemy czterech rodzajów punktów (dane przykładowe):

match, +1 punkt, np. A z A, G z G, itd.

mismatch, -1 punkt, np. A z G

gap_{open}, g_{open} = -2 punkty – „otwierający” znak -, np. wiersze 3 i 4 z pierwszej kolumny, ale też 2ga kolumna wiersza w2 oraz 6 kolumna wiersza w4

gap_{extend}, g_{ext} = -1 punkt, w przykładzie to druga, trzecia i czwarta kolumna wiersza w3

Ok, ale jak oceniamy? Ocena za całe dopasowanie to suma ocen częściowych każdy-z-każdym, określa ich liczbę dwumian Newtona, w naszym przypadku „4 na 2”, gdzie 4 to liczba wierszy. Jakbyśmy mieli nie cztery ale 14 sekwencji do dopasowania, to liczba porównań to byłoby „14 na 2”.

Mniejsza z dwumianem i jego wynikiem, i tak ma to robić komputer. W naszym przypadku musimy najpierw ocenić dopasowania w1-w2, potem w1-w3, w1-w4, następnie: w2-w3, w2-w4 i finalnie w3-w4.

Ocena za dopasowania w1 z w2: (A G T C G T A G oraz A - T C G T C G)

match A-A: +1

g_{open} G i „-” : -2 punkty

match T-T: +1

match C-C: +1

match G-G: +1

match T-T: +1

mismatch A-C: -1

match G-G: +1

Czyli suma w1-w2: +1 -2 +1 +1 +1 +1 -1 +1 = 3. Tego typu „ambitne” obliczenia musi wykonać komputer dla każdego porównania. Czyli jest pięć zostało. Zróbmy jeszcze w1-w3:

g_{open} A -: -2 punkty

g_{ext} G -: -1 punkt

g_{ext} T -: -1

g_{ext} C -: -1

match G-G : +1

match T-T : +1

match A-A : +1

match G-G : +1

Suma w1-w3: -2 -1 -1 -1 +1 +1 +1 +1 = -5 + 4 = -1

Kolejne: suma w1-w4: 0, w2-w3: -3, w2-w4: -5, w3-w4: -3

Ocena całkowita dopasowania: 3 - 1 + 0 - 3 - 5 - 3 = -9

Czy to dużo czy to mało? Nie wiemy, możemy co najwyżej wymyślić inne dopasowania i porównywać wyniki. Naszym celem jest oczywiście **maksymalizacja**, czyli najlepsze dopasowania to te z największą liczbą, nawet gdyby była ujemna.

Spójrzmy jeszcze raz na dopasowanie:

w1	A	G	T	C	G	T	A	G
w2	A	-	T	C	G	T	C	G
w3	-	-	-	-	G	T	A	G
w4	-	G	T	A	G	-	A	G

Macierz binarna:

1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	1	1	1	1	0	1	1	1

Czyli nasze dopasowanie jest opisane macierzą w której jedynka oznacza literę, a zero oznacza kreskę. Tylko tyle i aż tyle. Ale uwaga, z tego wynika sposób generowania rozwiązań losowych:

- musimy wygenerować tyle wierszy ile jest sekwencji do dopasowania
- wiersz opisujący daną sekwencję wejściową musi mieć **dokładnie tyle jedynek, ile dane sekwencja ma liter.**

Kolumny w tej macierzy możemy sobie nazywać jak chcemy, np. **ruchami**. W tym znaczenie **ruch** to coś, co prowadzi nas dalej, do konstrukcji rozwiązania dopuszczalnego. **Rozwiązanie dopuszczalne** jest właśnie zdefiniowane tymi dwoma powyższymi punktami nad tym akapitem – tyle wierszy ile sekwencji + liczba jedynek w wierszu = liczba liter w danej sekwencji (reszta to zera).

Należy też zauważyć, że każdy ruch można oceniać niezależnie. Np. pierwsza kolumna to „1 1 0 0”. Jest tam +1 punkt za match nukleotydów A z pierwszego i drugiego wiersza. A także cztery razy po -2 punkty za „gap_{open}” (wiersz w1 z w3 (-2 pkt.) i i w4 (-2 pkt.), oraz wiersz w2 z w3 (-2 pkt) i w2 z w4 (-2). Razem: +1 – 2 – 2 – 2 – 2 = -7.

W drugiej kolumnie „1 0 0 1” mamy dwa raz „gap_{open}” (w1 z w2 oraz w2 z w4, każdy karany -2 pkt.), dwa razy „gap_{extend}” (w1 z w3 oraz w3 z w4, każde karane -1 pkt), 1 match (w1 z w4, +1 pkt). Porównanie gap’ów (w2 z w3) jest za zero punktów. Czyli +1 – 2 – 2 – 1 – 1 + 0 = +1 – 6 = -5. Każdy ruch/kolumnę można więc oceniać niezależnie kolumna po kolumnie, tak jak na poprzedniej stronie było pokazane, jak wiersz po wierszu liczyliśmy funkcję celu/jakość dopasowania.