

Dla Kade

‘‘Wstępow do obliczeniowej biologii molekularnej’’
(J. Tiuryn, wykład nr.6, 23 listopada 2005)

Spis treści

3 Przeszukiwanie baz danych	36
3.1 Heurystyczne algorytmy	36
3.1.1 FASTA	36
3.1.2 BLAST	38
3.1.3 Statystyka porównywania sekwencji	39
3.1.4 PSI-BLAST	41
3.1.5 Uliniowanie sekwencji z profilem	42
3.2 Macierze substytucyjne	44
3.2.1 PAM	44
3.2.2 BLOSUM	46

3 Przeszukiwanie baz danych

Jest dużo baz danych specjalizujących się w różnych aspektach związanych z sekwencjami naturalnie pojawiającymi się w biologii. Dla sekwencji DNA główne bazy danych to: GenBank (USA), EMBL (Europa), DDBJ (Japonia). Natomiast dla białek główną bazą danych jest Swiss-Prot. Przeszukiwanie baz danych jest jedną z głównych metod pracy współczesnego biologa. Poniżej omówimy kilka zagadnień związanych z praktycznymi aspektami przeszukiwania baz.

3.1 Heurystyczne algorytmy

Omówimy dwa najbardziej popularne heurystyczne algorytmy używane do obliczania przybliżonej wartości lokalnego uliniowania. Algorytmy te stanowią standardowe narzędzie do przeszukiwania baz danych w procesie wyszukiwania podobieństw pomiędzy sekwencjami.

3.1.1 FASTA

Jest to heurystyczny algorytm (Lipman, Pearson, 1985) służący do przybliżonego obliczania lokalnego uliniowania danego wzorca Q względem tekstowej sekwencji T wziętej z bazy danych. Zwykle stosuje się go do kolejnych sekwencji z bazy danych. Na początku użytkownik wybiera liczbowy parametr, zwany *ktup*. Standardowo sugerowane wartości dla *ktup* to 6 dla

sekwencji DNA oraz 2 dla białek. Przyjmijmy, że k jest wartością parametru *ktup*. Przez k -słowo będziemy rozumieć dowolne słowo długości k . Niech $n = |Q|$ oraz $m = |T|$. Działanie algorytmu można przedstawić w następujących czterech krokach.

1. Dla $1 \leq i \leq n$, $1 \leq j \leq m$ algorytm znajduje pary (i, j) , takie że k -słowo zaczynające się w Q w pozycji i jest identyczne z k -słowem zaczynającym się w T w pozycji j . Każda taka para (i, j) nazywa się *gorącym miejscem*. Operację tę można wykonać efektywnie sporządzając na początku tablicę haszującą dla Q , lub (rzadziej) dla wszystkich słów T z bazy danych.
2. Każde gorące miejsce (i, j) można traktować jako odcinek długości k leżący na przekątnej o numerze¹ $(i - j)$ w tablicy V (otrzymanej metodą dynamicznego programowania — V oczywiście nie mamy). Algorytm przypisuje pewne wartości dodatnie gorącym miejscom oraz wartości ujemne przerwom pomiędzy takimi miejscami (im dłuższa przerwa, tym mniejsza wartość). Dla każdej przekątnej zawierającej gorące miejsca, algorytm wybiera fragment pomiędzy gorącymi miejscami o maksymalnej wartości. W ten sposób zostaje wybranych 10 przekątnych (i zawartych w nich fragmentów) o maksymalnej wartości. Dla każdego z tych fragmentów algorytm znajduje część takiego fragmentu (podśłowo) o maksymalnej wartości uliniowienia bez spacji (do obliczania tej wartości stosuje się tablicę PAM lub BLOSUM) Taką część fragmentu nazwiemy *poduliniowieniem*. Niech *init1* będzie najlepszym poduliniowieniem.
3. Wybrane są poduliniowienia, których wartość przekracza pewną z góry ustaloną granicę. Z tych dobrych poduliniowień próbuje się ułożyć uliniowienie o maksymalnej wartości. W tym celu buduje się następujący graf poduliniowień. Wierzchołkami są poduliniowienia. Każdemu wierzchołkowi jest przypisana liczba będąca wartością tego poduliniowienia. Jeśli u i v są poduliniowieniami, takimi że u zaczyna się w pozycji (i, j) i kończy w pozycji $(i + d, j + d)$, a v zaczyna się w pozycji (i', j') , to tworzymy krawędź od u do v , gdy $i' > i + d$ oraz $j' > j + d$, tzn. gdy wiersz (kolumna) w którym zaczyna się v jest poniżej wiersza (na prawo od kolumny), w którym kończy się u . Krawędzi tej przypisujemy pewną wagę zależącą od liczby spacji jakie trzeba wprowadzić we fragmencie lokalnego uliniowienia, w którym poduliniowienie v występuje po poduliniowieniu u . Im większa liczba spacji tym waga takiej krawędzi jest

¹Główna przekątna ma numer 0, przekątne o numerach dodatnich leżą nad główną przekątną, a o numerach ujemnych pod główną przekątną.

mniejsza. Następnie algorytm znajduje drogę o maksymalnej wartości w wyżej opisanym grafie. Taka droga wyznacza lokalne uliniowanie pomiędzy dwoma słowami. To nie musi być optymalne lokalne uliniowanie pomiędzy Q oraz T . Oznaczmy to uliniowanie przez $initn$.

4. Algorytm wraca do poduliniowania $init1$ z kroku 2 i znajduje najlepsze lokalne uliniowanie wokół przekątnej zawierającej $init1$ w pasie $[-8, 8]$ (dla białek) oraz w pasie $[-16, 16]$ (dla DNA). Niech opt będzie takim uliniowaniem.

W ten sposób Q jest porównywane z kolejnymi słowami T z bazy danych. Biorąc pod uwagę opt lub $initn$ wyznacza się małą liczbę słów T , najbardziej obiecujących z punktu widzenia uliniowania z Q . Dla każdego z nich wykonuje się pełny algorytm Smitha-Watermana obliczający optymalne uliniowania.

3.1.2 BLAST

Algorytm BLAST (Altschul *et.al.* 1990) podaje jako wynik całe spektrum rozwiązań (uliniowień) wraz z oszacowaniem statystycznej istotności znalezionej wartości (czyli prawdopodobieństwa tego, że znaleziona wartość, lub wartość od niej większa mogła się pojawić przypadkiem (z losowej sekwencji)).

BLAST porównuje wzorzec Q z każdą sekwencją z bazy danych, starając się zidentyfikować te sekwencje T , dla których MSP (*maximal segment pair*, czyli para podśłów równej długości maksymalizująca wartość uliniowania bez spacji²) jest większe od pewnej z góry ustalonej wartości C . W ten sposób wybiera się pewne słowa T “podejrzane” o pewne podobieństwo z Q .

Jak się szuka takich T , dla których MSP jest większe od C ? Ustala się długość w oraz wartość graniczną t . Następnie BLAST znajduje wszystkie w -podśłowa w T , dla których istnieje w -podśłowo w Q o wartości uliniowania (bez spacji) większej od t . Każde takie miejsce jest rozszerzane w celu znalezienia wartości uliniowania większej od C . Jeśli w trakcie rozszerzania wartość uliniowania (która może rosnać lub maleć z każdym krokiem rozszerzenia) spadnie poniżej pewnej wartości progowej, to poszukiwania dla takiego miejsca są przerywane.

Dobór wartości C , w oraz t ma kluczowe znaczenie dla jakości znajdowanych wyników. Na przykład, dla porównywania białek w jest przyjmowane pomiędzy 3 a 5, natomiast dla DNA jest zwykle równe około 12.

²Przy użyciu pewnej macierzy substytucyjnej.

3.1.3 Statystyka porównywania sekwencji

Poniżej przedstawimy podstawowe wyniki teorii (Karlin, Altschul'1990), na której BLAST opiera analizę statystycznej istotności znalezionej sekwencji uliniowania. Teoria ta nie dotyczy uliniowań ze spacjami. Opracowanie analogicznej teorii dla uliniowań ze spacjami stanowi problem otwarty.

Analiza jednej sekwencji

Na początek zajmijmy się analizą probabilistyczną jednej sekwencji. Dany jest alfabet $A = \{a_1, \dots, a_r\}$, z którego losowany jest ciąg liter o prawdopodobieństwach $\{p_1, \dots, p_r\}$. Z każdym wystąpieniem litery a_i w sekwencji związana jest wartość s_i będąca liczbą rzeczywistą. Przyjmujemy następujące założenia:

1. Dla pewnego i mamy $s_i > 0$.
2. Wartość oczekiwana wartości dla całego alfabetu jest ujemna: $\sum_{i=1}^r p_i s_i < 0$.

Rozważamy zmienną losową $M(n)$ przyjmującą wartość maksymalną dla segmentu w losowej sekwencji długości n .

Twierdzenie 3.1.1 *Wartość oczekiwana dla $M(n)$ jest rzędu $\frac{\ln n}{\lambda^*}$, gdzie λ^* jest jedynym dodatnim rozwiązaniem równania*

$$\sum_{i=1}^r p_i \cdot e^{\lambda \cdot s_i} = 1.$$

Twierdzenie 3.1.2

$$\mathbf{Prob} \left(M(n) - \frac{\ln n}{\lambda^*} > x \right) \approx 1 - e^{-K \cdot e^{-\lambda^* x}},$$

gdzie K jest stałą zadaną szybko zbieżnym szeregiem.

Zatem wycentrowana zmienna losowa $\tilde{M} = m(n) - \frac{\ln n}{\lambda^*}$ ma rozkład EVD (extreme value distribution), zwany też rozkładem Gumbela.

Ponadto oczekiwana liczba wystąpień segmentów w losowej sekwencji długości n , o wartości większej niż $\frac{\ln n}{\lambda^*} + x$ wynosi $K \cdot e^{-\lambda^* x}$.

Powyższe twierdzenie wynika z następującej ogólniejszej uwagi: liczba wystąpień 'oddzielnych' segmentów o wysokiej wartości (tzn. o wartości większej niż $\frac{\ln n}{\lambda^*} + x$) jest aproksymowane przez rozkład Poissona o parametrze $a = K \cdot e^{-\lambda^* x}$.

Przypomnijmy, że rozkład Poissona o parametrze a dla zmiennej losowej x przyjmującej wartości naturalne wygląda następująco

$$\mathbf{Prob}(X = k) = \frac{a^k}{k!} \cdot e^{-a}.$$

Zatem prawdopodobieństwo napotkania m lub więcej różnych segmentów o dużej wartości wynosi $1 - e^{-a} \cdot \sum_{i=1}^{m-1} \frac{a^i}{i!}$. Przyjmując $m = 1$ dostajemy pierwszą część twierdzenia. Druga część wynika natychmiast z faktu, że wartość oczekiwana dla zmiennej losowej o rozkładzie Poissona z parametrem a wynosi $E(X) = a$.

Niech $S = \frac{\ln n}{\lambda^*} + x$ będzie wartością segmentu w losowej sekwencji długości n . Wówczas, zgodnie z Twierdzeniem 3.1.2, oczekiwana liczba wystąpień segmentów rozłącznych o wartości co najmniej S wynosi $K \cdot e^{-\lambda^* x} = K \cdot e^{\ln n - \lambda^* S} = K \cdot n \cdot e^{-\lambda^* S}$. To jest właśnie tzw. E-value obliczane przez program BLAST.

$$E = K \cdot n \cdot e^{\lambda^* S}.$$

Statystyka porównywania dwóch sekwencji

Mamy dwie sekwencje: jedna losowana z rozkładem na literach $\{p_1, \dots, p_r\}$, a druga z rozkładem $\{p'_1, \dots, p'_r\}$. Ponadto mamy tablicę substytucyjną $(s_{i,j})_{1 \leq i,j \leq r}$. Przyjmujemy następujące założenia:

1. $\sum_{i,j} p_i p'_j s_{i,j} < 0$, oraz
2. Dla pewnych i, j , mamy $s_{i,j} > 0$.
3. Rozkłady $\{p_1, \dots, p_r\}$ oraz $\{p'_1, \dots, p'_r\}$ nie różnią się zbyttnio od siebie.³

Dalej analiza wygląda podobnie do przypadku jednej sekwencji o długości $m \cdot n$, gdzie m i n są długościami losowych sekwencji. W szczególności niech λ^* będzie jedynym dodatnim rozwiązaniem równania $\sum_{i,j} p_i p'_j e^{\lambda^* s_{i,j}} = 1$. Niech $M(m, n)$ będzie zmienną losową wyrażającą maksymalną wartość lokalnego uliniowienia (bez spacji) losowych słów o długościach m oraz n generowanych z powyższych rozkładów. Wówczas mamy następujące twierdzenie

$$\mathbf{Prob} \left(M(m, n) > \frac{\ln mn}{\lambda^*} + x \right) \approx 1 - e^{-K \cdot e^{-\lambda^* x}}.$$

Jest to tzw. wartość p -value, czyli prawdopodobieństwo tego, że można uliniować parę losowych sekwencji o długościach m oraz n tak, że natrafimy na

³Techniczną definicję tego założenia pomijamy tutaj, odsyłając zainteresowanego czytelnika do publikacji.

segment uliniowienia (bez spacji) o wartości przekraczającej $\frac{\ln mn}{\lambda^*} + x$. Ponadto oczekiwana liczba segmentów o wartości przekraczającej $S = \frac{\ln mn}{\lambda^*} + x$ wynosi $E = K \cdot e^{-\lambda^* x} = K \cdot m \cdot n \cdot e^{-\lambda^* S}$. Jest tzw. E-value. Program BLAST używa wartości E-value zamiast p-value ze względu na większy zakres wartości przyjmowanych w przypadku E-value. Zauważmy, że dla małych wartości E-value (np. dla $E < 0.01$) obie wartości są niemal identyczne.

Wartości stałych K oraz λ^* zależą od tablicy substytucyjnej oraz od rozkładów na literach. Przykładowo, dla tablicy BLOSUM62 oraz rozkładzie na aminokwasach obliczonym dla danych zawartych w bazie danych mamy $\lambda^* = 0.3176$ oraz $K = 0.134$.

Obserwowaną wartość S zwykle zastępuje się tzw. wartością *znormalizowaną* S' (mierzoną w bitach), czyli taką wartością aby zachodziła zależność

$$E = \frac{m \cdot n}{2^{S'}}.$$

Łatwo jest policzyć, że wówczas mamy

$$S' = \frac{\lambda^* S - \ln K}{\ln 2}.$$

Tak więc znormalizowana wartość lokalnego uliniowienia oraz E-value są ze sobą ściśle związane i związek ten zależy jedynie od rozmiaru obydwu sekwencji. Przykładowo dla sekwencji długości 250 oraz bazy danych zawierającej 50 milionów znaków E-value 0.005 odpowiada znormalizowanej wartości uliniowienia około 38 bitów.

3.1.4 PSI-BLAST

Program PSI-BLAST (Position-Specific Iterated BLAST), a właściwie rodzina programów, powstał w 1997r. Zawiera on szereg ulepszeń w stosunku do starszej wersji. Trzy główne ulepszenia wymienione są poniżej.

1. Poszukiwanie par ‘hitów’ na przekątnych, zamiast pojedynczych ‘hitów’. Prowadzi to do większej wrażliwości.
2. Dopuszczenie przerw w uliniowaniach oraz statystyczna analiza dla przerw. Nie ma tutaj wspierającej teorii, ale symulacje komputerowe pokazują, że rozkłady mają podobny kształt jak w przypadku uliniowań bez przerw. Stałe λ^* oraz K wylicza się eksperymentalnie (symulacje stochastyczne).
3. Iteracyjnie konstruuje się profil (inaczej PSSM - Position-Specific Score Matrix) i przeszukuje bazę danych uliniawiając profil z kolejnymi sek-

wencjami (por. rozdział 3.1.5 poświęcony uliniowieniu profilu a sekwencją). Profil, na danym kroku, powstaje przez wzięcie wszystkich istotnych uliniowień z wrorcem (z poprzedniego kroku). Takie wielokrotne uliniowanie wyznacza nowy profil, który staje się nowym wzorcem. Iteracja kończy się po zadanej liczbie kroków. Szerokość profilu jest równa długości początkowego wzorca.

Jako przykład zastosowania programu PSI-BLAST autorzy podają eksperyment biorąc jako wzorec ludzkie białko HIT do przeszukania bazy danych SWISS-PROT. Znaleziono zostały (jako statystycznie istotne) tylko inne białka z rodziny HIT o wartości E-value mniejszej niż 0.01. Porównując struktury stwierdzono, że HIT jest istotnie podobne do białka GalT (ze szczura). Wartość E-value z BLAST'a dla tych dwóch białek wynosi 0.012, a więc nieco poniżej wyżej wymienionej wartości progowej. Po pierwszej iteracji PSI-BLAST wartość E-value dla GalT (ze szczura) spadła do $2 \cdot 10^{-4}$, a ponadto odkryto inne białko GalT (z bakterii *H. influenzae*) o wartości E-value równej $4 \cdot 10^{-5}$. Po drugiej iteracji odkryto podobieństwo o wartości E-value równej $2 \cdot 10^{-4}$ do pewnego białka z drożdży, dla którego struktura nie jest znana.

3.1.5 Uliniowanie sekwencji z profilem

Przypuśćmy, że mamy rodzinę sekwencji uliniowaną w następujący sposób:

```
a b c - a
a b a b a
a c c b -
c b - b c
```

Powyższe ulinowanie generuje profil w następujący sposób. Dla każdej kolumny i każdego symbolu (również spacji) liczymy częstość wystąpienia tego symbolu w kolumnie. W ten sposób powstaje tablica $P = (p_{x,i})$, gdzie x przebiega po wszystkich symbolach, a i przebiega po numerach kolumn. Ta tablica nazywa się *profilem*. W naszym przykładzie mamy następujący profil:

	1	2	3	4	5
a	0.75	0	0.25	0	0.50
b	0	0.75	0	0.75	0
c	0.25	0.25	0.50	0	0.25
-	0	0	0.25	0.25	0.25

Tak więc profil jest skończonym ciągiem rozkładów prawdopodobieństwa na zbiorze symboli $\Sigma \cup \{-\}$. Mając dane słowo $S \in \Sigma^*$ oraz profil P , *ulinowienie* S z P jest to każde ustawienie symboli z S z kolumnami z P (z możliwością wstawiania spacji zarówno pomiędzy symbole z S , jak i pomiędzy kolumny z P), przy którym spacja nie stoi na wprost spacji. Przyjmujemy, że mamy rozkład prawdopodobieństwa p_- odpowiadający spacji wstawianej do sekwencji profilu. Rozkład ten jest skupiony na symbolu $-$, tzn. $p_-(-) = 1$ oraz $p_-(x) = 0$ dla $x \neq -$. Przykładem uliniowienia dla powyższego profilu P ze słowem $aabbc$ jest

a a b - b c
1 - 2 3 4 5

Zatem w ogólności uliniowaniem słowa S z profilem P jest para słów $S^\# \in (\Sigma \cup \{-\})^*$, $P^\# \in (\mathbb{N} \cup \{-\})^*$, taka że

1. $|S^\#| = |P^\#|$.
2. $S^\#$, po usunięciu spacji daje S .
3. $P^\#$, po usunięciu spacji, daje ciąg $1 \dots n$.
4. Dla każdego i , $S^\#(i)$ oraz $P^\#(i)$ nie są jednocześnie spacją.

Niech $s : (\Sigma \cup \{-\})^2 \rightarrow \mathbb{R}$ będzie funkcją podobieństwa. Wartością uliniowania $(S^\#, P^\#)$ jest liczba

$$\sum_{i=1}^{|S^\#|} s(S^\#(i), P^\#(i)),$$

gdzie dla $x \in \Sigma \cup \{-\}$ oraz dla rozkładu p_i (gdzie $i \in \{1, \dots, n\} \cup \{-\}$) występującego w profilu P ,

$$s(x, i) = \sum_{y \in \Sigma \cup \{-\}} s(x, y) \cdot p_i(y).$$

W naszym przykładzie, jeśli funkcją podobieństwa jest

$$s(x, y) = \begin{cases} 1 & \text{jeśli } x = y, \\ -1 & \text{jeśli } x \neq y, \end{cases}$$

to wartość uliniowania przedstawionego powyżej wynosi $0.5 - 1 + 0.5 - 0.5 + 0.5 - 0.5 = -.05$.

Problem znajdowania optymalnego uliniowania słowa z profilem rozwiązuje się, używając metody dynamicznego programowania, w taki sam sposób jak dla przypadku dwóch słów. Jedyna różnica polega na tym, że obecnie mamy do czynienia z dwoma alfabetami, nad którymi są brane słowa: Σ oraz zbiór rozkładów prawdopodobieństw nad $\Sigma \cup \{-\}$. Ponieważ koszt obliczania wartości podobieństwa $s(x, i)$ wynosi $O(|\Sigma|)$ (przyjmujemy koszt liczenia $s(x, y)$ oraz $p_i(y)$ jako jednostkowe), to całkowity czas, w którym znajdujemy optymalne uliniowanie słowa S z profilem P wynosi $O(|\Sigma| \cdot m \cdot n)$, gdzie $m = |S|$, $n = |P|$ (tzn. P jest ciągiem n rozkładów).

3.2 Macierze substytucyjne

Zacznijmy od paru ogólnych uwag. Załóżmy, że dla liter $a, b \in \Sigma$, $q_{a,b}$ jest prawdopodobieństwem tego, że aminokwasy a oraz b pochodzą od wspólnego aminokwasu (przodka) w drodze punktowej mutacji. Mamy dane dwa słowa $x = x_1 \dots x_n$ oraz $y = y_1 \dots y_n$. Chcemy obliczyć jakie jest prawdopodobieństwo tego że x oraz y pochodzą od wspólnego przodka w wyniku punktowych zmian. Dla dowolnej litery a , niech p_a oznacza prawdopodobieństwo wystąpienia litery a w słowie. Wówczas prawdopodobieństwo pojawienia się słów x oraz y to

$$\prod_{i=1}^n p_{x_i} \cdot \prod_{i=1}^n p_{y_i}.$$

Natomiast prawdopodobieństwo tego, że x i y pochodzą od wspólnego przodka jest równe $\prod_{i=1}^n q_{x_i, y_i}$. Iloraz tych dwóch wartości

$$\prod_{i=1}^n \frac{q_{x_i, y_i}}{p_{x_i} \cdot p_{y_i}}$$

nazywa się *odds ratio*. Im większa jest ta wartość tym bardziej dane dwa słowa x, y nie są całkowicie losowe (niezależne), ale pochodzą od wspólnego przodka. Ponieważ dużo wygodniej jest pracować z addytywną miarą podobieństwa, to przechodzimy do logarytmu:

$$s(x, y) = \sum_{i=1}^n \log\left(\frac{q_{x_i, y_i}}{p_{x_i} \cdot p_{y_i}}\right),$$

liczba

$$s(a, b) = \log\left(\frac{q_{a,b}}{p_a \cdot p_b}\right)$$

jest karą/nagrodą za zamianę a na b .

3.2.1 PAM

Macierze PAM (*percent accepted mutations*) zostały zaproponowane przez M. Dayhoff i jej współpracowników około 1978r. Są to tzw. macierze substytucyjne dla aminokwasów i jako takie reprezentują funkcję podobieństwa. PAM również używa się jako jednostki miary opisującej ewolucyjną rozbieżność pomiędzy dwoma ciągami aminokwasów. Powiemy, że dwa słowa S_1 i S_2 różnią się o jedną jednoskę PAM, jeśli S_2 można otrzymać z S_1 w ciągu akceptowalnych punktowych mutacji, tak że średnia liczba akceptowalnych mutacji na 100 aminokwasów wynosi 1. Akceptowalna punktowa mutacja to taka, która albo nie zmieniła funkcji białka, lub była korzystna dla organizmu (co najmniej nie spowodowała śmierci organizmu). Zwróćmy uwagę, że różnica 1PAM pomiędzy S_1 i S_2 nie oznacza, że słowa te różnią się pomiędzy sobą o 1%. Liczba mutacji na pewnej pozycji w słowach może być większa od jedynki, a nawet w wyniku tych mutacji ta pozycja nie musi się zmienić.

Jako podstawę do budowy macierzy PAM wzięto pewną rodzinę bardzo podobnych białek (każde dwa nie różniły się o więcej niż 15%) i ręcznie sporządzono globalne uliniowienia dla tej rodziny. Następnie stworzono tablicę A , taką że dla aminokwasów a, b , $A(a, b)$ jest liczbą wystąpień uliniowienia pary (a, b) w wyżej wymienionych uliniowieniach. Wówczas prawdopodobieństwo mutacji z a na b jest równe

$$q_{a,b} = \frac{A(a,b)}{\sum_c A(a,c)}.$$

Niech p_a będzie prawdopodobieństwem wystąpienia litery a w w/w białkach. Wówczas wartość oczekiwana (średnia) liczby zamian w losowej parze słów długości m wynosi

$$m \cdot \sum_{a,b} p_a \cdot p_b \cdot q_{a,b}.$$

Ponieważ macierz 1PAM to taka, dla której powyższa wartość oczekiwana wynosi 1 dla $m = 100$, to musimy tak przeskalować $q_{a,b}$ na $q'_{a,b}$ aby zachodziło

$$\sum_{a,b} p_a \cdot p_b \cdot q'_{a,b} = 0.01.$$

Wystarczy stosownie dobrać stałą d i wziąć:

$$q'_{a,b} = \begin{cases} d \cdot q_{a,b} & \text{dla } a \neq b, \\ d \cdot q_{a,b} + (1-d) & \text{dla } a = b. \end{cases}$$

W ten sposób otrzymujemy 1PAM macierz. Oznaczmy ją przez $S(1)$. Tak więc (w przybliżeniu) $S(1)(a, b)$ jest prawdopodobieństwem zamiany a na b w

jednej umownej jednostce czasu. Macierz $S(1)$ przedstawia pewien łańcuch Markowa. Prawdopodobieństwo zamiany a na b w n jednostkach czasu to $S(1)^n(a, b)$, gdzie $S(1)^n$ jest n -krotnym iloczynem macierzy $S(1)$ przez siebie. Macierze $S(1)^n$ nazywają się n PAM macierzami. Wartości do prawdziwej macierzy n PAM są brane z $S(1)^n$ przez stosowanie logarytmu, skalowanie i zaokrąglanie. Bardzo popularne są 250PAM macierze.

3.2.2 BLOSUM

Macierze substytucyjne BLOSUM (blocks substitution matrix) zostały zaproponowane przez S. oraz J. Heinkoff w 1991r. jako efekt krytyki tego, że macierze n PAM dla $n \gg 1$ nie oddają dobrze wpływu czasu na zmiany sekwencji kodujących białka. Macierze BLOSUM zostały oparte na białkowej bazie danych BLOCKS w następujący sposób. Niech $0 \leq L \leq 100$. Opiszemy sposób budowania macierzy BLOSUM L . Białka z bazy danych są dzielone na grupy. Do jednej grupy są zaliczane dwa białka jeśli można przejść od jednego białka do drugiego, używając białek pośrednich wziętych z bazy danych, takich że każde dwa kolejne białka mają skład identyczny w co najmniej $L\%$. Białka w bazie danych BLOCKS są uliniowane w tzw. blokach. Tworzymy macierz A . Wybierzmy dwie grupy g, g' . Dla aminokwasów a, b , liczba $A^{g,g'}(a, b)$ jest częstością z jaką aminokwas a pochodzący z grupy g jest uliniowany z aminokwasem b pochodzącym z grupy g' (liczba ta jest podzielona przez $n \cdot n'$, gdzie n jest licznością grupy g , a n' jest licznością grupy g'). $A(a, b)$ otrzymuje się przez sumowanie $A^{g,g'}(a, b)$ po wszystkich parach grup g, g' . Mając A możemy obliczyć prawdopodobieństwo wystąpienia aminokwasu a :

$$p_a = \frac{\sum_b A(a, b)}{\sum_{c,d} A(c, d)}$$

oraz prawdopodobieństwo zamiany a na b :

$$q_{a,b} = \frac{A(a, b)}{\sum_{c,d} A(c, d)}.$$

Wówczas

$$s(a, b) = \log\left(\frac{q_{a,b}}{p_a \cdot p_b}\right).$$

Najczęściej używane L to $L = 50$ oraz $L = 62$.