Liniowe dopasowanie dwóch sekwencji II

Dopasowanie lokalne

Przypomnienie. Dysponujemy funkcją podobieństwa liter $s:(\Sigma')^2\to R$, gdzie $\Sigma'=\Sigma\cup\{-\}$. Z jej pomocą określamy podobieństwo liniowego dopasowania u^*,w^* słów $u,w\in\Sigma^*$:

$$ş(u^*,w^*)=\Sigma_{i=1,\ldots,|u^*|}\, s(u^*[i],w^*[i])$$

oraz wartość podobieństwa samych słów:

$$ş(u,w)=\max_{(u^*,w^*)\in Dopasowania(u,w)} ş(u^*,w^*)$$

Optymalne dopasowanie można znaleźć w czasie $O(|u||w|)$.

W praktyce rzadko dysponujemy dwoma precyzyjnie zlokalizowanymi istotnymi sekwencjami. Zamiast tego posiadamy długi zsekwencjonowany fragment, w którym należy znaleźć odcinki zbliżone do sekwencji o znanych funkcjach biologicznych (np. geny, promotory, regulatory).

Problem lokalnego dopasowania. Dane są słowa $u,w\in\Sigma^+$, $|u|=n$, $|w|=m$. Należy znaleźć ich podsłowa (odpowiednio u',w') o maksymalnej możliwej wartości dopasowania $ş(u',w')$:

$$H(u,w)=\max_{i,j,k,l} ş(u[i..j],w[k..l])$$

Przykład. Słowa u=’pqraxabcstvq’, w=’xyaxbacsll’, zaś $s(a,a)=2$, $s(a,b)=-2$, $s(a,'-')=s('-',a)=-1$ dla $a\neq b\in\Sigma$.
Optymalne lokalne dopasowanie ma postać
u'=axab–cs
w'=ax–bacs
Wartość podobieństwa 8.

Jak to rozwiązać?
Pierwszy sposób: dla wszystkich par podsłów z u ($n(n+1)/2+1$) i w ($m(m+1)/2+1$) wyznaczamy wartość podobieństwa. Czas $O(\max\{n,m\}^6)$ – za długo!
Drugi sposób: algorytm optymalnego dopasowania określał podobieństwa dla wszystkich przedrostków badanych słów, wystarczy więc sprawdzać pary przyrostków z u i w (jest ich $\Theta(nm)$). Czas $O(\max\{n,m\}^4)$.

Jeszcze szybciej?

Algorytm.

Tworzymy tablicę
$v(i,j)$=maksymalne możliwe podobieństwo jakiegoś przyrostka $u[1..i]$ z przyrostkiem z $w[1..j]$ (uwzględniamy tu nawet ε).

1. Brzeg tabeli: $v(0,i)=v(i,0)=0$

2. Środek tabeli ($i,j>0$):
$v(i,j)=\max\{0, v(i-1,j-1)+s(u[i],w[j]), v(i,j-1)+s('-',w[j]), v(i-1,j)+s(u[i],'-')\}$.

3. $H(u,w)$ jest maksymalną wartością w tabeli $v[0..n,0..m]$.

Złożoności czasowa i pamięciowa $O(nm)$.

Jak znaleźć podsłowa?

- w 2. za każdym razem gdy 0 nie jest maksimum, wstawiamy odpowiedni wskaźnik $(i,j)\rightarrow$ $(i–1,j–1)$ lub $(i,j–1)$ lub $(i–1,j)$.
- w 3. niech $v(i',j')$ będzie największą wartością w tabeli v, wówczas dowolna maksymalna ścieżka wychodząca z $v(i',j')$ określa podsłowa i ich optymalne dopasowanie.

Przykład. Jak znaleźć dwa nie zachodzące na siebie podsłowa o maksymalnym możliwym podobieństwie w sekwencji $v=a_1...a_n$?

Algorytm. (czas $O(n^3)$)
M=0;
for i=1 **to** n–1 **do** M=max(M,$H(a_1...a_i, a_{i+1}...a_n)$);

Dopasowanie map restrykcyjnych

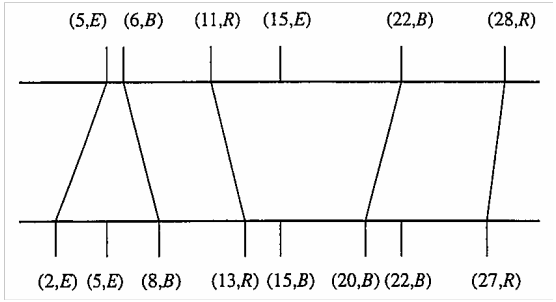
Enzymy restrykcyjne:

- jedno z najważniejszych narzędzi do obróbki nici DNA,
- rozcina nić w miejscach wystąpienia sekwencji charakterystycznej dla enzymu,
- istnieją enzymy o rozmaitych „częstościach cięcia", rozdrabniające DNA na krótsze lub dłuższe fragmenty.

Dla odcinka DNA (przed zsekwencjonowaniem) często tworzy się mapy określające położenie (np. w parach zasad od końca nici) punktów charakterystycznych. Pozwala to na opis rozmieszczenia interesujących nas fragmentów (np. konkretnych genów). Podobną postać mogą mieć mapy oparte na wykrywaniu unikalnych znaczników (np. Sequence Tagged Sites).

Przyjmijmy, że mapa nici (o pewnej znanej długości) ma postać ciągu $(s_1,d_1),...,(s_n,d_n)$, gdzie s_i – klucz określający rodzaj i–tego punktu charakterystycznego (np. typ enzymu restrykcyjnego), d_i – odległość od początku nici ($d_1<d_2<...<d_n$).

- Mapy tego rodzaju zawierają przybliżone dane.
- Chcemy określić podobieństwa dwóch nici DNA porównując ich mapy.



Przykład dopasowywania map.

Określamy **odległość edycyjną między mapami** *MD* jako minimalny koszt ciągu operacji przekształcających pierwszą mapę w drugą (stadia pośrednie muszą być poprawnymi mapami). Dopuszczalne jest:

- wstawienie lub usunięcie pary (s_i,d_i) – koszt λ
- indel odcinka długości t nie zawierającego punktu mapowania (przesuwają się wówczas położone dalej miejsca (s_i,d_i)) – koszt βt.

Algorytm. Dane: dwie mapy dla nici o znanych długościach $A=(s_1,d_1),...,(s_n,d_n)$, $B=(s'_1,d'_1),...,(s'_m,d'_m)$

1. Dodajemy do obu map „nowy początek": $(s_0,0)$ oraz nowe końce (s,dl_A), (s, dl_B) – s_0 i s są unikalne.

Niech $MD(i,j)$ – minimalny koszt konwersji podmap $A_i=(s_0,0),...,(s_i,d_i)$ z $B_j=(s_0,0),...,(s'_j,d'_j)$, przy czym ostatni punkt A_i przechodzi na koniec B_j.

2. Inicjacja tabeli. Dla $i\in\{1,...,n\}$, $j\in\{1,...,m\}$:
$MD(0,0)=0$;
$MD(0,j)= MD(n+1,j)= MD(i,0)= MD(i,m+1)= \infty$;

$MD(i,j)= \infty$ dla wszystkich $s_i\neq s'_j$ – typy punktów cięcia muszą zgadzać się ze sobą.

3. Jeżeli $s_i=s'_j$: badamy możliwe wcześniejsze punkty map A_i i B_j pod kątem ewentualnej zgodności:

$$MD(i,j)=\min_{0<k\le i,\ 0<l\le j} [MD(i-k,j-l)+\lambda(l+k-2)+ +\beta|(d_i-d_{i-k})-(d'_j-d'_{j-l})|]$$

4. Odczytujemy odległość edycyjną $MD(A,B)= MD(n+1,m+1)$

Dopasowanie słowa i grafu

Dany jest digraf acykliczny $D(V,E)$ z wyróżnionymi wierzchołkami Z (źródło) i U (ujście) oraz łukami poetykietowanymi literami $l:E\to\Sigma$. Ponadto znamy słowo $w\in\Sigma^*$. Szukamy ścieżki p z Z do U, takiej że konkatenacja u_p liter jej kolejnych łuków ma maksymalną możliwą wartość podobieństwa $ş(w,u_p)$ z w.

Algorytm.

1. Numerujemy wierzchołki z V „topologicznie" kolejnymi liczbami naturalnymi (istnienie łuku $v_i\to v_j$ implikuje $i<j$).

Niech $SD(i,v)$ dla $i\le|w|$ i $v\in V$ oznacza maksymalne podobieństwo prefiksu $w[1..i]$ ze słowem pochodzącym ze ścieżki $Z\to v$.

2. Wyznaczamy $SD(0,v)$ – długość najdłuższej ścieżki $Z\to v$, przy czym krawędzi e przypisujemy długość $s(l(e),'-')$.

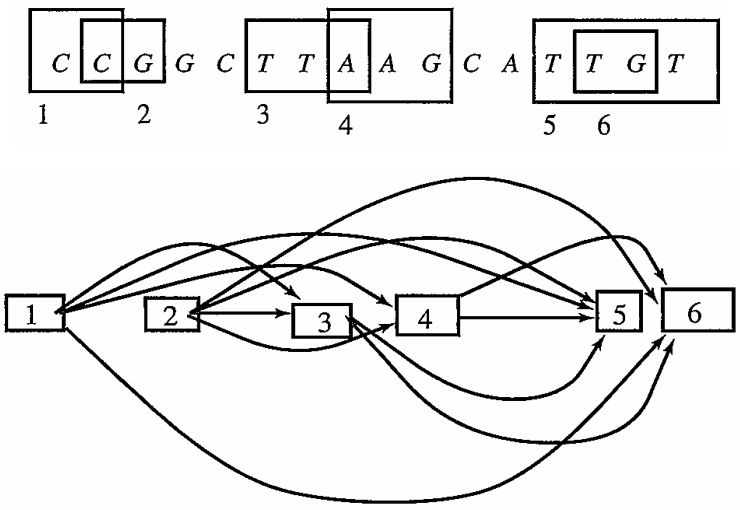
3. Dla kolejnych $i=1,...,|w|$ i wierzchołków v (wg rosnącej numeracji) obliczamy:

$$SD(i,v)=\max\{\max_{(u,v)\in E} SD(i-1,u)+s(w[i],l((u,v))),\ \max_{(u,v)\in E} SD(i,u)+s('-',l((u,v))),\ SD(i-1,v)+s(w[i],'-')\}.$$

4. Odczytujemy rozwiązanie $SD(|w|,U)$.

Czas $O(|w||E|)$.

Zastosowanie: **gene prediction problem**.
Dysponujemy zsekwencjonowanym odcinkiem DNA, który podejrzewamy o zawieranie jakiegoś genu. Nie wiemy gdzie leżą eksony i introny. Istnieją heurystyczne procedury pozwalające na namierzenie odcinków „prawdopodobnie będących eksonami" (np. długie **otwarte ramki odczytu** tj. bez kodonu STOP, odpowiednie częstości występowania pewnych kodonów itp.) – są one jednak zawodne, a wynikowe bloki często zachodzą na siebie.



Konstrukcja sieci bloków.

1. Tworzymy sieć bloków – łuk od bloku A do B oznacza, że koniec A w sekwencji poprzedza początek B.
2. Każdy blok zastępujemy ścieżką z krawędziami etykietowanymi jego kolejnymi literami, łuki łączące bloki obciążamy „sztuczną” literą o zerowym koszcie usunięcia.
3. Dodajemy *Z* i *U*.
4. Dopasowujemy sieć ze znanymi sekwencjami aminokwasów z bazy białek (np. w postaci cDNA tj. po odwrotnej transkrypcji na nukleotydy) – homologiczny gen pozwoli na wyznaczenie kolejnych nie zachodzących na siebie eksonów.

Skąd wziąć funkcję podobieństwa *s*?

Rozważamy słowa $w=a_1...a_n$ i $v=b_1...b_n$ dopasowane bez przerw. Przyjmujemy model niezależnej ewolucji poszczególnych miejsc (nie do końca prawda, por. wyspy CpG). Znając większy zbiór poprawnie wielodopasowanych sekwencji (np. homologiczne wersje tego samego enzymu u różnych organizmów) możemy oszacować prawdopodobieństwo $q(a,b)$ zamiany a z b, $a,b\in\Sigma$. Podobnie $p(a)$ – częstość występowania litery a w typowych słowach.

model A – obie sekwencje są niezależne.
model B – są ze sobą „spokrewnione”.

$\Pr(w,v|A)=\Pi_{i=1,...,n}\, p(a_i)\, \Pi_{j=1,...,n}\, p(b_j)$
$\Pr(w,v|B)=\Pi_{i=1,...,n}\, q(a_i,b_i)$

Chcielibyśmy, aby dodatnia wartość podobieństwa $S(w,v)>0$ wykrywała fakt, że $\Pr(w,v|B)>\Pr(w,v|A)$, czyli że $\Pr(w,v|B)/\Pr(w,v|A)>1$. Po zlogarytmowaniu:

$$\log[\Pr(w,v|B)/\Pr(w,v|A)]= \Sigma_{i=1,...,n} \log[q(a_i,b_i)/\, p(a_i)p(b_i)]>0$$

Dlatego dogodnie jest obrać:

$$s(a,b)= \log[q(a,b)/\, p(a)p(b)] \text{ dla } a,b\in\Sigma.$$

Trudniej jest wyznaczyć kary za przerwy ...

Macierz PAM: uzyskana na podstawie analizy sekwencji podobnych białek macierz (20×20) prawdopodobieństwa zamian aminokwasów w umownej jednostce czasu, w której średnio wymianie ulega 1% pozycji. Podstawą jest macierz 1PAM, w praktyce używa się analogicznych macierzy dla dłuższych okresów np. 50PAM, 250PAM.

Obecnie PAM wypierają macierze **BLOSUM** (np. BLOSUM50) utworzone na podstawie bazy danych dopasowanych do siebie sekwencji podobnych enzymów.

Modelowanie ewolucji sekwencji

Rozważamy fragmenty dopasowane bez przerw. Można przyjąć założenie upraszczające o niezależnej ewolucji na poszczególnych pozycjach. Litera jest przy tym podejściu stanem **łańcucha Markowa**. Niech $|\Sigma|=n$. Przyjmijmy, że znamy macierz ($n\times n$) przejść $U(1)$ zachodzących w umownej jednostce czasu ($U(1)_{ab}$ jest prawdopodobieństwem zastąpienia litery b przez a w tym czasie). Po dwóch jednostkach czasu mamy podobną macierz:

$$U(2)_{ab} = \Sigma_{c\in\Sigma}\, U(1)_{ac}\, U(1)_{cb},$$

czyli w zapisie macierzowym $U(2)=U(1)^2$. Ogólniej po czasie n macierz przejść ma postać:

$$U(n)=U(1)^n.$$

Rachunek ten przeprowadzono dla macierzy PAM (Σ – aminokwasy).

Możemy też zastosować łańcucha Markowa z ciągłym czasem. Rozważamy zmiany $p_i(t)$ prawdopodobieństwa i–tej litery:

$$\mathrm{d}p_i(t)/\mathrm{dt} = [\Sigma_{j\in\{1,\ldots,n\}-\{i\}} R_{ij}\, p_j(t)] + R_{ii}\, p_i(t)$$

gdzie:
R_{ij} – średnia „krótkookresowa" szybkość zamian $j\to i$.
$R_{ii} = -\Sigma_{j\in\{1,\ldots,n\}-\{i\}} R_{ji}$ – średnia szybkość „ubywania" stanu i.

Macierzowo: niech $P(t)$ – kolumna prawdopodobieństw $p_i(t)$.

$$\mathrm{d}P(\mathrm{t})/\mathrm{dt} = R\, P(t)$$

– przyjmujemy początkowy rozkład $P(0)=P_0$.

Znane jest rozwiązanie ogólne równania różniczkowego:

$$P(t)=e^{Rt}\, P_0,$$

– gdzie dla macierzy B definiujemy $e^B=\Sigma_{k=0,\ldots,\infty} B^k/k!$

Zatem $U(t)= e^{Rt}$ – macierz przejść po czasie t.

Przykład. Najprostszy **model Jukes – Cantor** ewolucji DNA. Każda zamiana zachodzi z jednakową częstością α. Macierz R ma postać:

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix} \end{array}$$

Rozwiązaniem jest macierz prawdopodobieństw przejść po czasie t:

$$U(t)=\begin{bmatrix} r_t & s_t & s_t & s_t \\ s_t & r_t & s_t & s_t \\ s_t & s_t & r_t & s_t \\ s_t & s_t & s_t & r_t \end{bmatrix}$$

gdzie:

$$r_t=(1+3e^{-4\alpha t})/4, \qquad s_t=(1-e^{-4\alpha t})/4.$$

Własności:

- każda zamiana $a\to b$ ($a\neq b$) jest jednakowo prawdopodobna (pr. wynosi s_t),
- bez względu na literę początkową dla $t\to\infty$ każda z 4 zasad pojawi się z prawdopodobieństwem ¼,
- oczekiwana liczba „obserwowanych" substytucji po czasie t wyniesie $3s_t$.

Ostatni wzór pozwala oszacować czas t oddzielenia się obu sekwencji na podstawie średniej obserwowanej częstości podstawień f:

$$t=-\ln(1-4f/3)/4\alpha$$

Proporcjonalny do t jest tzw. **Jukes–Cantor distance** między sekwencjami:

$$d^{JC}=3\alpha t=-(3/4)\ln(1-4f/3)$$

W chwili początkowej (t=0) nici DNA są identyczne (f=0), więc i odległość d^{JC}=0. Gdy $t\to\infty$ ilość obserwowanych substytucji dąży do f=3/4 całego ciągu, więc $d^{JC}\to\infty$.

Przykład. <u>Model Kimury</u> – **<u>tranzycje</u>** (zamiana puryna ↔ puryna lub pirymidyna ↔ pirymidyna) zwykle zachodzą częściej niż **<u>transwersje</u>** (puryna ↔ pirymidyna). W niektórych genach mitochodrialnych różnica jest nawet kilkunastokrotna! Kimura wprowadza osobno: częstość tranzycji α i częstość transwersji β. Macierz R:

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array}\begin{bmatrix} -2\beta-\alpha & \beta & \alpha & \beta \\ \beta & -2\beta-\alpha & \beta & \alpha \\ \alpha & \beta & -2\beta-\alpha & \beta \\ \beta & \alpha & \beta & -2\beta-\alpha \end{bmatrix}$$

Prawdopodobieństwa przejść po czasie t:

$$U(t)=\begin{bmatrix} r_t & s_t & u_t & s_t \\ s_t & r_t & s_t & u_t \\ u_t & s_t & r_t & s_t \\ s_t & u_t & s_t & r_t \end{bmatrix}$$

gdzie:

$$s_t=(1-e^{-4\beta t})/4,\ u_t=(1+e^{-4\beta t}-2e^{-2(\alpha+\beta)t})/4,\ r_t=1-2s_t-u_t.$$

Własności:

- prawdopodobieństwo zaobserwowania zamiany $a\to b$ ($a\neq b$) po czasie t wynosi u_t dla tranzycji i s_t dla transwersji,
- bez względu na literę początkową przy $t\to\infty$ każda z 4 zasad pojawi się z prawdopodobieństwem ¼,
- oczekiwana częstość „obserwowanych" tranzycji po czasie t wyniesie $P=u_t$, a transwersji $Q=2s_t$.

Kimura distance między sekwencjami:

$$d^K=(\alpha+2\beta)t=-\ln(1-2P-Q)/2-\ln(1-2Q)/4$$

nie takie proste ...

- Szybkość mutacji jest różna u różnych organizmów.

- Szybkość zmian sekwencji białek jest bardzo różnorodna – znamy struktury silnie „konserwatywne” (np. histony) jak i szybko zmienne.

- Odcinki kodujące DNA zmieniają się wolniej niż „śmieciowe” – jest to związane z presją darwinowską ograniczającą swobodę mutacji istotnych fragmentów.

- W obszarze kodującym będą częściej występować nie wpływające na sekwencje białek mutacje synonimiczne. Z tego samego powodu w obrębie kodonów pozycja trzecia zwykle zmienia się częściej (czasem nawet kilkakrotnie) niż druga. Porównanie częstości zmian w zależności od położenia wewnątrz kodonu pozwala na określenie rodzaju presji darwinowskiej wywoływanej przez dobór naturalny na konkretne białko i szybkość jego ewolucji.

- Oczywiście jeśli dwie sekwencje miały ostatniego wspólnego przodka t lat temu, dzieli je dystans czasowy niezależnej ewolucji równy $2t$.