

‘‘Wstę do obliczeniowej biologii molekularnej’’
(J. Tiuryn, wykłady nr. 12 i 13; 25 stycznia 2006)

Spis treści

8	Konstrukcja drzew filogenetycznych	82
8.1	Metoda UPGMA	82
8.2	Metoda łączenia sąsiadów	84
8.3	Metoda parsymonii	87

8 Konstrukcja drzew filogenetycznych

Drzewa filogenetyczne przedstawiają historię ewolucji gatunków. Historia ta jest przedstawiona w postaci binarnego drzewa. Długości krawędzi w takim drzewie odpowiadają ilości czasu jaki upłynął pomiędzy zdarzeniami ewolucyjnymi. Liście są etykietowane nazwami organizmów, a wierzchołki wewnętrzne przedstawiają zdarzenia ewolucyjne zwane *specjacją* – jest to sytuacja, gdy z jednego gatunku, w drodze procesów ewolucyjnych powstają dwa gatunki. Tak więc korzeń takiego drzewa odpowiada praprzodkowi wszystkich gatunków rozważanych w drzewie filogenetycznym. Jako podstawę do budowy drzewa filogenetycznego zwykle przyjmuje się rodzinę genów, po jednym z każdego organizmu, o których sądzimy, że wszystkie pochodzą od wspólnego przodka (genu) w drodze specjacji, czyli że są *ortologami*.

Omówimy tu trzy podejścia do konstrukcji drzew filogenetycznych: dwa oparte na metodach metrycznych (UPGMA oraz metoda łączenia sąsiadów) oraz metodę parsymoniczną (MP).

8.1 Metoda UPGMA

Nazwa tej metody pochodzi od *Unweighted Pair Group using arithmetic Averages*. Przypuśćmy, że mamy n sekwencji S_1, \dots, S_n oraz dla każdej pary sekwencji S_i, S_j (dla $i \neq j$) mamy obliczoną miarę $d_{i,j}$ przedstawiającą odległość ewolucyjną pomiędzy tymi sekwencjami. Miarę tę uogólnimy na klasy w następujący sposób. Niech C, C' będą rozłącznymi zbiorami sekwencji. Definiujemy

$$d_{C,C'} = \frac{1}{|C| \cdot |C'|} \cdot \sum_{p \in C, q \in C'} d_{pq}. \quad (8.1)$$

Jest to średnia arytmetyczna odległości pomiędzy elementami klastrów C i C' . Z tego względu ta metoda bywa nazywana również *average linkage clustering*, w odróżnieniu od *complete linkage clustering* (gdy zamiast średniej bierzemy maksimum odległości), oraz od *single linkage clustering* (gdy zamiast średniej bierzemy minimum odległości).

Zauważmy, że jeśli $C_1 \cap C_2 = \emptyset$ oraz $C \cap (C_1 \cup C_2) = \emptyset$, to mamy

$$\begin{aligned}
 d_{C_1 \cup C_2, C} &= \frac{1}{|C_1 \cup C_2| \cdot |C|} \cdot \sum_{p \in C_1 \cup C_2, q \in C} d_{pq} \\
 &= \frac{1}{|C_1 \cup C_2| \cdot |C|} \cdot \left(\sum_{p \in C_1, q \in C} d_{pq} + \sum_{p \in C_2, q \in C} d_{pq} \right) \\
 &= \frac{1}{|C_1 \cup C_2| \cdot |C|} \cdot (|C_1| \cdot |C| \cdot d_{C_1, C} + |C_2| \cdot |C| \cdot d_{C_2, C}) \\
 &= \frac{|C_1| \cdot d_{C_1, C} + |C_2| \cdot d_{C_2, C}}{|C_1| + |C_2|}. \tag{8.2}
 \end{aligned}$$

Prowadzi to do następującego algorytmu:

Algorytm UPGMA

Dane: n sekwencji S_1, \dots, S_n oraz macierz odległości $\{d_{ij}\}$ dla $i, j = 1, \dots, n$. Iteracyjnie modyfikujemy listę L klastrów oraz drzew, po jednym drzewie związanym z każdym klastrem z L . Początkowo lista L składa się z jednoelementowych klastrów $C_i = \{i\}$. Klastrowi C_i odpowiada jednowierzchołkowe drzewo, którego wierzchołek jest etykietowany przez i .

while $|L| > 1$ **do**

wybierz dwa klastry $C_1, C_2 \in L$ takie, że d_{C_1, C_2} jest minimalne (jeśli jest więcej niż jedna para to wybieramy dowolną). Usuń z L klastry C_1 i C_2 oraz dodaj nowy klastrowi $C_1 \cup C_2$. Odległość pomiędzy nowym klastrem a pozostałymi na liście definiujemy według wzoru (8.2). Drzewo związane z klastrem $C_1 \cup C_2$ powstaje przez wzięcie drzew dla klastrów C_1 i C_2 , i połączenie ich przez dodanie korzenia na wysokości $\frac{1}{2} \cdot d_{C_1, C_2}$ od poziomemu liści.

od

Wyjście: drzewo odpowiadające jednemu klastrowi jaki pozostał na liście w chwili zakończenia pracy.

Można udowodnić, że powyższa konstrukcja jest poprawna w następującym sensie

Lemat 8.1.1 *Na każdym kroku konstrukcji drzewa przez algorytm UPGMA, wysokość (względem poziomu liści) dodawanego nowego wierzchołka jest nie mniejsza od wysokości jego synów.*

Pozostaje do rozstrzygnięcia jeszcze jeden problem. Otóż okazuje się, że nie zawsze tak musi być, że odległość pomiędzy liśćmi odczytana z drzewa skonstruowanego metodą UPGMA pokrywa się z odległościami odczytanymi z macierzy $\{d_{ij}\}$. Zjawisko to jest związane z tzw. *hipotezą zegara molekularnego*, która zakłada że zmiany sekwencji zachodzą z tą samą prędkością we wszystkich punktach drzewa. Zachodzenie tej hipotezy oznacza, że dla każdego wierzchołka drzewa długość ścieżki do każdego liścia poniżej tego wierzchołka jest taka sama. Powiemy, że macierz $\{d_{ij}\}$ spełnia warunek *ultrametryczności*, gdy dla każdej trójki i, j, k , odległości d_{ij}, d_{jk}, d_{ik} są albo wszystkie równe, lub dwie są sobie równe i większe od trzeciej.

Lemat 8.1.2 *Odległości odczytane z drzewa skonstruowanego metodą UPGMA pokrywają się z macierzą $\{d_{ij}\}$ wtedy i tylko wtedy, gdy ta macierz spełnia warunek ultrametryczności.*

Zadanie 8.1.1 Udowodnić Lemat 8.1.1. Podać przykład danych, dla których drzewo konstruowane metodą UPGMA nie jest binarne.

Zadanie 8.1.2 Udowodnić Lemat 8.1.2.

8.2 Metoda łączenia sąsiadów

Niech T będzie drzewem o n liściach i o długościach przypisanych wszystkim krawędziom. Niech $\{d_{ij}\}$ będzie macierzą odległości pomiędzy liśćmi. Załóżmy, że $\{d_{ij}\}$ spełnia aksjomaty metryki, tzn.

- $d_{ij} \geq 0$ oraz $(d_{ij} = 0 \iff i = j)$.
- $d_{ij} = d_{ji}$.
- $d_{ij} \leq d_{ik} + d_{kj}$.

Powiemy, że T jest *addytywne* dla $\{d_{ij}\}$, gdy dla dowolnych i, j , d_{ij} jest równe sumie długości wszystkich krawędzi leżących na drodze w T od i do j ¹

Zajmiemy się obecnie metodą odtwarzania T na podstawie macierzy $\{d_{ij}\}$, przy założeniu, że drzewo T jest addytywne. Powiemy, że dwa liście i, j w drzewie T są *sąsiadami*, gdy i oraz j mają wspólnego rodzica. Zauważmy,

¹Suma po pustym zbiorze jest równa 0.

że jeśli i oraz j są liśćmi będącymi sąsiadami w T o wspólnym wierzchołku (rodzicu) v , to dla każdego liścia $m \neq i, j$ mamy

$$d_{vm} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij}), \quad (8.3)$$

gdzie d_{vm} jest odległością w T od v do m .

Pozostaje do rozstrzygnięcia kwestia jak rozpoznawać sąsiadów na podstawie macierzy odległości $\{d_{ij}\}$. Zauważmy, że minimalność d_{ij} nie gwarantuje, że para i, j jest sąsiadami. Pomysł tego podejścia polega na umiejętnym poprawieniu macierzy $\{d_{ij}\}$.

Założmy, że $n \geq 3$. Dla dowolnego liścia i definiujemy

$$r_i = \frac{1}{n-2} \cdot \sum_{k=1}^n d_{ki}. \quad (8.4)$$

Twierdzenie 8.2.1 (Studier, Keppeler (1988))

Niech T będzie drzewem o co najmniej trzech liściach, addytywnym dla $\{d_{ij}\}$. Każda para i, j , która minimalizuje

$$\rho_{ij} = d_{ij} - (r_i + r_j) \quad (8.5)$$

jest parą sąsiadów.

Uwaga: liczby ρ_{ij} mogą być ujemne.

Zadanie 8.2.1 Udowodnić Twierdzenie 8.2.1. Podać przykład, że twierdzenie to przestaje być prawdziwe, gdy we wzorze (8.4) zastąpimy $n-2$ przez $n-1$ lub n .

Zadanie 8.2.2 Niech liczba liści n będzie równa 3. Podać warunki konieczne i dostateczne na $\{d_{ij}\}$, przy których istnieje drzewo addytywne dla $\{d_{ij}\}$.

Algorytm ‘łączenie sąsiadów’ (NJ)

Dane: n sekwencji ($n \geq 3$) S_1, \dots, S_n oraz macierz odległości $\{d_{ij}\}$.

Iteracyjnie modyfikujemy listę L wierzchołków wraz z odległościami pomiędzy nimi. Każdemu wierzchołkowi $v \in L$ odpowiada drzewo, którego korzeniem jest v . Początkowo $L = \{1, \dots, n\}$ składa się z wszystkich liści oraz liściowi i odpowiada jednowierzchołkowe drzewo o korzeniu i .

while $|L| > 2$ **do**

Wybierz dowolną parę $i, j \in L$, dla której ρ_{ij} (por. (8.5)) jest minimalne. Utwórz nowy wierzchołek v , dodaj go do L oraz usuń z L wierzchołki i, j . Dla

$m \in L - \{i, j\}$ definiujemy $d_{vm} = \frac{1}{2} \cdot (d_{iv} + d_{jm} - d_{ij})$. Drzewo odpowiadające v powstaje z drzew odpowiadających i oraz j przez wstawienie korzenia v i dodanie krawędzi z v do i o długości $d_{iv} = \frac{1}{2} \cdot (d_{ij} + r_i - r_j)$ oraz dodanie krawędzi z v do j o długości $d_{jv} = \frac{1}{2} \cdot (d_{ij} + r_j - r_i)$.

od

Jeśli L zawiera tylko dwa wierzchołki u, v o odpowiadających im drzewach T_u i T_v , to tworzymy nieukorzenione drzewo T przez połączenie v z u krawędzią o długości d_{uv} .

Poprawność definicji d_{vm} w powyższym algorytmie wynika z (8.3) oraz z Twierdzenia 8.2.1. Poprawność definicji d_{iv} oraz d_{jv} wynika z następującego rozumowania. Zauważmy, że jeśli i oraz j są sąsiadami, to dla każdego liścia $m \neq i, j$ mamy

$$d_{iv} = \frac{1}{2} \cdot (d_{ij} + d_{im} - d_{jm}).$$

Zatem

$$(n-2) \cdot d_{iv} = \frac{1}{2} \sum_{m \neq i, j} (d_{ij} + d_{im} - d_{jm}) = \frac{1}{2} \cdot ((n-2) \cdot d_{ij} + \sum_m d_{im} - \sum_m d_{jm}).$$

Stąd dostajemy

$$d_{iv} = \frac{1}{2} \cdot (d_{ij} + r_i - r_j).$$

Uwaga: konstrukcja drzewa metodą łączenia sąsiadów nie podaje położenia korzenia.

Na koniec podamy warunki charakteryzujące to czy dana metryka $\{d_{ij}\}$ pochodzi od drzewa addytywnego względem $\{d_{ij}\}$.

Twierdzenie 8.2.2 (Buneman'1971)

Niech $\{d_{ij}\}$ będzie metryką na $\{1, \dots, n\}$. Następujące warunki są równoważne:

- (i) Istnieje drzewo T addytywne dla $\{d_{ij}\}$.
- (ii) $\{d_{ij}\}$ spełnia następujący **warunek czterech punktów**: dla dowolnych czterech różnych elementów i, j, k, l dwie wartości spośród

$$d_{ij} + d_{kl}, \quad d_{ik} + d_{jl}, \quad d_{il} + d_{jk}$$

są sobie równe i większe od tej trzeciej.

Zadanie 8.2.3 Udowodnić Twierdzenie 8.2.2.

Zadanie 8.2.4 Czy dana metryka $\{d_{ij}\}$ może mieć dwa różne drzewa addytywne?

Zadanie 8.2.5 Pokazać, że jeśli $\{d_{ij}\}$ ma drzewo addytywne, to algorytm łączenia sąsiadów znajdzie to drzewo.

Zadanie 8.2.6 Pokazać, że jeśli $\{d_{ij}\}$ spełnia warunek ultrametryczności to spełnia warunek czterech punktów. Czy odwrotna implikacja jest prawdziwa?

8.3 Metoda parsymonii

Metoda maksymalnej parsymonii (czyli oszczędności) polega na znalezieniu drzewa filogenetycznego, które wyjaśnia powstanie danych sekwencji w drodze ewolucji zawierającej minimalną liczbę podstawień. Problem znajdowania optymalnego drzewa rozбивa się na następujące dwa podproblemy:

- (P1) Obliczyć koszt dla danego drzewa T ;
- (P2) Przeszukać przestrzeń wszystkich drzew, aby znaleźć drzewo o minimalnym koszcie.

Zajmiemy się najpierw problemem pierwszym. Zakładamy, że mamy n sekwencji S_1, \dots, S_n , wszystkie o tej samej długości m . O tych sekwencjach możemy myśleć tak, że powstały przez uliniowanie wyjściowych sekwencji S'_1, \dots, S'_n (niekoniecznie o tej samej długości). Ponieważ liczbę podstawień oblicza się niezależnie dla każdej pozycji $i = 1, \dots, m$, to wystarczy zająć się obliczeniem kosztu dla zadanego drzewa T , zakładając, że w liściach T stoją pojedyncze symbole alfabetu. Przyjmujemy tu oczywiście, że T ma n liści, i -ty liść etykietowany symbolem a_i .

Rozwiążemy to zadanie dla nieco uogólnionego problemu, tzw. *ważonej parsymonii*. Przyjmujemy, że mamy daną funkcję kosztu $S : \Sigma^2 \rightarrow \mathbb{R}$, która każdej parze symboli a, b przypisuje koszt $S(a, b)$ podstawienia a za b . Zastosujemy metodę programowania dynamicznego. Dla $a \in \Sigma$ oraz wierzchołka v drzewa T definiujemy $V(v, a)$ jako minimalny koszt poddrzewa zaczepionego w v przy założeniu, że etykietą wierzchołka v jest a . Mamy następujące warunki dla liścia v , $V(v, a) = 0$, jeśli etykietą v w T jest a . W przeciwnym przypadku przyjmujemy $V(v, a) = +\infty$.

Jeśli v nie jest liściem oraz i, j są synami v , to

$$V(v, a) = \min_b [V(i, b) + S(a, b)] + \min_b [V(j, b) + S(a, b)].$$

Wówczas koszt dla całego drzewa T wynosi $\min_b V(v_*, b)$, gdzie v_* jest korzeniem T .

Złożoność czasowa powyższego algorytmu wynosi $O(|T| \cdot |\Sigma|^2)$, a przestrzenna $O(|T| \cdot |\Sigma|)$. Odtwarzanie etykiet wierzchołków wewnętrznych otrzymuje się standardową metodą wskaźników.

Klasyczne podejście parsymoniczne otrzymujemy przyjmując $S(a, b) = 1$ dla $a \neq b$, oraz $S(a, a) = 0$. Tradycyjny algorytm Fitcha (1971) obliczania kosztu wygląda następująco. Zaczynając od liści, konstruujemy dla każdego wierzchołka v zbiór R_v potencjalnych etykiet realizujących minimalny koszt dla tego wierzchołka. W tablicy $C(v)$ będziemy zapamiętywać ten koszt. Poniższy algorytm stosuje metodę dynamicznego programowania.

Algorytm Fitcha

(Obliczanie R_v i $C(v)$)

Jeśli v jest liściem o etykiecie a , to $R_v := \{a\}$; $C(v) = 0$;

W przeciwnym przypadku, niech w, u będą synami v . Wówczas

Jeśli $R_w \cap R_u \neq \emptyset$, to $R_v := R_w \cap R_u$; $C(v) := C(w) + C(u)$;

W przeciwnym przypadku $R_v := R_w \cup R_u$; $C(v) := C(w) + C(u) + 1$;

Wyjście: minimalny koszt drzewa: $C(v_*)$, gdzie v_* jest korzeniem.

Jeśli chodzi o problem (P2), przeszukiwania przestrzeni wszystkich drzew w poszukiwaniu drzewa optymalnego, to stosuje się tutaj jedynie algorytmy heurystyczne. Jednym z nich jest tzw. metoda *branch and bound*. Metoda ta polega na systematycznym budowaniu drzew zaczynając od drzew o jednym liściu i przechodząc do drzew o coraz większej liczbie liści. Ponieważ koszt całego drzewa nie jest mniejszy od kosztu każdego z jego poddrzew, to przeszukiwanie przestrzeni drzew w danym kierunku przerywamy, gdy osiągniemy koszt większy od dotąd osiągniętego w poprzednich krokach. Skuteczność tej metody polega na wyborze odpowiedniego sposobu przechodzenia przestrzeni drzew.

Zadanie 8.3.1 Dowieść, że koszt drzewa (rozumiany tak jak w tej sekcji) nie zależy od położenia korzenia w tym drzewie.