‘‘Algorytmy biologii molekularnej’’
(J. Tiuryn, w18.tex, 2 maja 2000)

Spis rzeczy

9 Rearanżacje genomów – odległość inwersyjna 130

9 Rearanżacje genomów – odległość inwersyjna

W odróżnieniu od punktowych mutacji, mutacje na dużą skalę (na poziomie chromosomów lub całych genomów) pozwalają lepiej badać podobieństwa/różnice pomiędzy różnymi gatunkami, a co za tym idzie odtwarzać historię ich ewolucji. Podstawowymi zjawiskami leżącymi u podstaw ewolucji genomów są:

- *Duplikacja z modyfikacjami*– części genomu są duplikowane, a następnie poddawane różnym modyfikacjom.
- *Inwersja* – fragment DNA jest odwrócony.
- *Translokacja* – końce dwóch chromosomów (telomery) są zamieniane.
- *Transpozycja* – dwa sąsiadujące fragmenty DNA są zamieniane miejscami.

Przykładowo, geny znajdujące się w pewnych częściach genomu bardzo rzadko podlegają mutacjom, natomiast porządek pomiędzy genami może być dużo częściej zmieniany (mitochondrium roślin lub chromosom X u ssaków).

Omówimy problemy pojawiające się przy badaniu liczby inwersji. Dana jest permutacja $\pi = (i_1, \ldots, i_n)$. *Inwersja* permutacji π polega na wybraniu pozycji $k < m$ $(1 \leq k, m \leq n)$ i przestawieniu fragmentu od i_k do i_m w odwrotnej kolejności. W wyniku dostajemy permutację

$$\pi' = (i_1, \ldots, i_{k-1}, i_m, i_{m-1}, \ldots, i_{k+1}, i_k, i_{m+1}, \ldots, i_n).$$

Problem *odległości inwersyjnej* polega na tym, że dana jest permutacja π, należy znaleźć najmniejszą liczbę inwersji, które przeprowadzają π w permutację identycznościową $id = (1, 2, \ldots, n)$.

Twierdzenie 9.0.1 (Caprara'97)
Problem odległości inwersyjnej jest NP-trudny.

Poniżej przedstawimy dwa algorytmy aproksymacyjne dla problemu odległości inwersyjnej. Zaczniemy od definicji przerwy. Dana jest permutacja $\pi = (i_1, \ldots, i_n)$. Przyjmiemy umowę, że i_0 zawsze oznacza 0 oraz i_{n+1} zawsze oznacza liczbę $n+1$. Niech $0 \leq j \leq n$. Powiemy, że π ma *przerwę* (breakpoint) pomiędzy i_j oraz i_{j+1}, gdy $|i_j - i_{j+1}| > 1$.

Przykład 9.0.1 Niech

$$\pi = (3, 2, 4, 5, 1).$$

Permutacja ta ma przerwy pomiędzy: 0 i 3, 2 i 4, 5 i 1 oraz pomiędzy 1 i 6. □

Niech $\phi(\pi)$ oznacza liczbę przerw permutacji π.

Lemat 9.0.2 *Odległość inwersyjna dla π jest co najmniej $\phi(\pi)/2$.*

Dowód: Jedna inwersja może zmniejszać liczbę przerw o co najwyżej 2. Ponieważ identyczność nie ma przerw, to liczba inwersji prowadzących od π do identyczności jest co najmniej $\phi(\pi)/2$. ■

Pasem w permutacji π nazwiemy każdy fragment o maksymalnej długości nie zawierający przerw. Pas nazwiemy *rosnącym*, gdy zawiera on co najmniej dwie liczby oraz liczby w nim znajdujące się tworzą rosnący ciąg. Pas nazwiemy *malejącym*, gdy liczby w nim znajdujące się tworzą malejący ciąg (malejący pas może zawierać tylko jedną liczbę). Zauważmy, że każdy pas jest albo rosnący albo malejący.

Lemat 9.0.3 *Jeśli $\pi \neq id$ oraz π nie zawiera pasa malejącego, to istnieje inwersja nie zwiększająca liczby przerw oraz wprowadzająca pas malejący.*

Dowód: π musi zawierać rosnący pas: $i_j, \ldots, i_k$. Wówczas pomiędzy i_{j-1} oraz i_j musi być przerwa. Podobnie pomiędzy i_k oraz i_{k+1}. Zatem inwersja tego pasa nie zwiększą liczby przerw oraz tworzy pas malejący. ■

Lemat 9.0.4 *Jeśli π zawiera malejący pas, to istnieje inwersja zmniejszająca liczbę przerw.*

Dowód: Niech i_k będzie najmniejszą liczbą zawartą w malejącym pasie. Zatem i_k stoi na prawym końcu tego pasa. Zauważmy, że $i_{k+1} \neq i_k + 1$ (bo w przeciwnym przypadku i_k należałoby do pasa rosnącego). Również $i_{k+1} \neq i_k - 1$ (bo w przeciwnym przypadku $i_k - 1$ należałoby do pasa malejącego, wbrew temu, że i_k jest najmniejszym elementem znajdującym się pasie malejącym). Zatem pomiędzy i_k oraz i_{k+1} jest przerwa. Niech $i_j = i_k - 1$. Podobnie rozumujemy pokazując, że pomiędzy i_j oraz i_{j+1} jest przerwa.

Jeśli i_j jest na prawo od i_k (tzn. $k < j$), to odwracamy fragment $i_{k+1}, \ldots, i_j$. Powoduje to zmniejszenie liczby przerw o co najmniej 1. Jeśli i_j jest na lewo od i_k, to odwracamy fragment od i_{j+1} do i_k. Znowu, liczba przerw zmniejsza się o co najmniej jedną. ■

Lematy 9.0.3 oraz 9.0.4 prowadzą do następującego algorytmu:

while (są przerwy) {
  if (jest malejący pas)
    then znajdź inwersję zmniejszającą liczbę przerw;
    else znajdź inwersję wprowadzającą pas malejący
      i nie zwiększającą liczby przerw }

Twierdzenie 9.0.5 *Powyższy algorytm sprowadza dowolną permutację π do permutacji identycznościowej, używając co najwyżej $2 \cdot \phi(\pi)$ inwersji. Zatem liczba inwersji znaleziona przez ten algorytm jest nie większa niż 4-krotna odległść inwersyjna.*

Powyższy algorytm można poprawić jeśli oprzemy się na następującym lemacie.

Lemat 9.0.6 *Niech π zawiera malejący pas i załóżmy, że każda inwersja zmniejszająca liczbę przerw w π daje permutację nie zawierającą malejącego pasa. Wówczas istnieje inwersja zmniejszająca liczbę przerw o 2.*

Dowód: Niech i_k będzie najmniejszą liczbą zawartą w pasie malejącycm oraz niech i_m będzie największą liczbą zawartą w pasie malejącycm. Niech $i_{k'} = i_k - 1$ oraz niech $i_{m'} = i_m + 1$. Stosując rozumowanie podobne do tego z Lematu 9.0.4 pokazujemy, że pomiędzy $i_{k'}$ oraz $i_{k'+1}$, pomiędzy i_k oraz i_{k+1}, pomiędzy i_{m-1} oraz i_m i pomiędzy $i_{m'-1}$ oraz $i_{m'}$ są przerwy. Zauważmy, że $i_{k'}$ musi leżeć na lewo od i_k, bo w przeciwnym przypadku inwersja fragmentu: $i_{k+1}, \ldots, i_{k'}$ pozostawi malejący pas (ten zawierający i_k), bez zwiększania

liczby przerw. Z analogicznego powodu $i_{m'}$ musi leżeć na prawo od i_m. Zauważmy też, że i_m musi leżeć pomiędzy $i_{k'}$ oraz i_k, bo w przeciwnym przypadku inwersja zbliżająca do siebie $i_{k'}$ oraz i_k (por. dowód Lematu 9.0.4) pozostawi pas malejący (ten zawierający i_m). Z tego samego powodu i_k musi leżeć pomiędzy i_m oraz $i_{m'}$. Tak więc kolejność (od lewej do prawej), w jakiej występują wyróżnione elementy jest następująca: $i_{k'}, i_m, i_k, i_{m'}$.

Twierdzimy, że

$$i_{m'} = i_{k+1} \tag{9.9.1}$$

Istotnie, gdyby przedział $i_k, \ldots, i_{m'}$ zawierał jakieś liczby to musi zawierać rosnący pas (nie może zawierać malejącego pasa, bo w p.p. inwersja $i_{k'+1}, \ldots, i_k$ zmniejesza liczbę przerw, pozostawiając pas malejący). Zatem inwersja $i_m, \ldots, i_{m'-1}$ zmniejsza liczbę przerw, wprowadzając pas malejący. Otrzymana sprzeczność dowodzi (9.9.1).

Podobnie się pokazuje, że $i_{k'} = i_{m-1}$. Obydwie równości dowodzą, że inwersja $i_m, \ldots, i_k$ zmniejsza liczbę przerw o 2. To kończy dowód lematu. ■

Lemat 9.0.6 prowadzi do następującego algorytmu.

while (są przerwy) {
  if (jest inwersja zmniejszająca liczbę przerw i pozostawiająca
      pas malejący)
    then wykonaj tę inwersję;
    else znajdź inwersję zmniejszającą liczbę przerw o 2 }

Twierdzenie 9.0.7 *Powyższy algorytm wykonuje co najwyżej $\phi(\pi)$ inwersji aby przekształcić π w permutację identycznościową. Zatem liczba inwersji znaleziona przez ten algorytm jest nie większa niż dwukrotna odległść inwersyjna.*

Istnieje algorytm o błędzie nie większym niż 1.75.

Rozważane też były *permutacje ze znakami*. Inwersje oprócz zmieniania kolejności zmieniają również znak przestawianych elementów na przeciwny. Na przykład, permutacja $(+5, -2, -4, +3, -1)$ po zastosowaniu inwersji trzech środkowych elementów daje $(+5, -3, +4, +2, -1)$. *Problem odległości inwersyjnej ze znakami* polega na znalezieniu dla danej permutacji π ze znakami najmniejszą liczbę inwersji, które przekształcą π w permutację $(+1, +2, \ldots, +n)$. Motywacją dla tego problemu jest to, że odwrócenie fragmentu podwójnej nici DNA jest związane z zamianą pojedynczych nitek tego fragmentu (tak aby zachować kierunek od 5' do 3').

Ten problem, w odróżnieniu od problemu odległości bez znaków, ma dokładne rozwiązanie w czasie wielomianowym. Pierwszy algorytm wielomianowy został znaleziony przez Hannenhalli i Pevznera w 1995r. Ulepszony algorytm działający w czasie $O(n^2)$ został podany przez Kaplana, Shamira i Tarjana w 1997r.