Laboratorium: Lista 4

Lokalizacja genów z wykorzystaniem testu Studenta

Statystyka w zastosowaniach

Sylwia Majchrowska
Matematyka

12 kwietnia 2016

Spis treści

1 Wstęp

Istnieje kilka różnorodnych metod lokalizacji genów, czyli sposobów na wskazanie takich miejsc w łańcuchu DNA, które istotnie wpływają na rozważane cechy. Aby zlokalizować gen odpowiedzialny za interesującą nas cechę posługujemy się tak zwanymi markerami molekularnymi, czyli fragmentami łańcucha DNA, których genotyp możemy ustalić eksperymentalnie. Ze względu na korelację markera (znajdującego się blisko szukanego genu) i genu wpływającego na cechę mamy możliwość detekcji genu. Zazwyczaj dana korelacja jest dość niska i znalezienie genu nie jest łatwe, dlatego też zwykle osobniki krzyżuje się (krzyżówka wsteczna) w taki sposób aby jeden z nich był homozygotą (aa lub AA) ze względu na badaną cechę - forma rodzicielska - a drugi heterozygotycznym potomkiem.

Podsumowując, dla każdego osobnika możemy podać ciąg genotypów (kodowanych na przykład jako 0 i 1) oraz wartość interesującej nas cechy. Poszczególne genotypy będą zmiennymi objaśniającymi, a cecha zmienną objaśnianą.

2 Zadanie pierwsze

Aby wygenerować macierz genotypów dla $n = 500$ osobników z krzyżówki wstecznej na 3 chromosomach o długości 150 cM i odstępie między sąsiednimi markerami $\Delta = 1$ cM posłużono się przykładowym kodem do generacji genotypów na jednym chromosomie krzyżówki wstecznej omawianym na wykładzie.

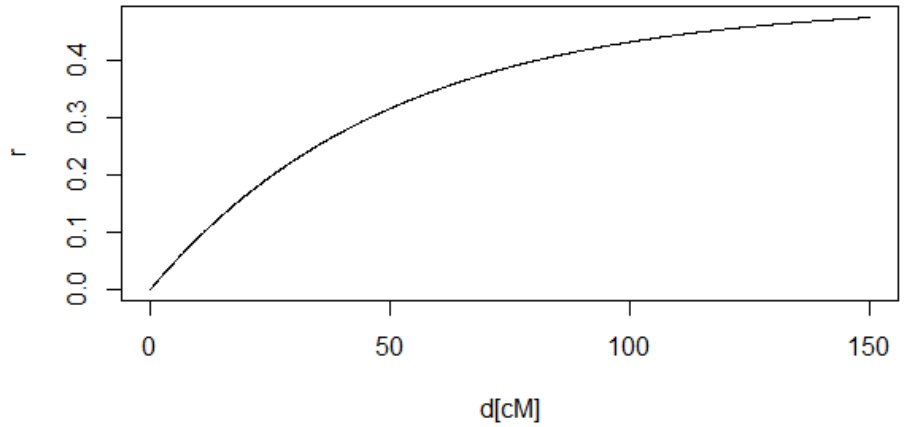
Oznaczmy przez M_i genotyp w i-tym markerze, taki, że

$$M_i = \begin{cases} 0 & \text{aa} \\ 1 & \text{Aa} \end{cases}$$

Między markerami doszło de rekombinacji (do crossing-over - skrzyżowania) jeżeli $M_i \neq M_j$, gdzie j oznacza kolejny marker. Mianem rekombinacji nazywamy nieparzystą ilość skrzyżowań. Dodatkowo zakładamy brak interferencji, czyli uznajemy, że zajście crossing-over w danym miejscu nie wpływa na prawdopodobieństwo zajścia kolejnego w pobliżu. Proces skrzyżowań modeluje się za pomocą procesu Poissona, gdzie czas oczekiwania na kolejne zdarzenie ma rozkład wykładniczy.

W takim przypadku zależność odległości genetycznej d, mierzonej w centymorganach (1 M to taka długość odcinka na chromosomie, że oczekiwana liczba skrzyżowań wynosi 1) od częstości rekombinacji wyrażana jest za pomocą funkcji Haldane'a

$$r(d) = \frac{1}{2}\left(1 - \exp(-0.02d)\right).$$



Rysunek 1: Funkcja mapowa Haldane'a. Ze wzrostem odległości częstość c-o dąży do 0.5.

Na potrzeby zadania utworzono trzy macierze X, Y oraz Z odpowiadające trójce chromosomów. Aby zweryfikować ich poprawność obliczono kolejno korelacje próbkowe między genotypami pierwszego i piątego markera i porównano je z wartością teoretyczną wyliczaną za pomocą wzoru

$$\rho(X_1, X_5) = 1 - 2r = \exp(-0.02d) = \exp(-0.02 * 4),$$

gdzie d oznacza odległość między sąsiednimi markerami. W naszym przypadku $d = 4\,\text{cM}$, ponieważ liczymy odległość między 1. a 5. markerem.

Wartości korelacji między genotypami pierwszego i piątego markera:

Chromosom	Korelacja próbkowa	Korelacja teoretyczna
X	0.9120873	
Y	0.9355299	0.9231163
Z	0.9199488	

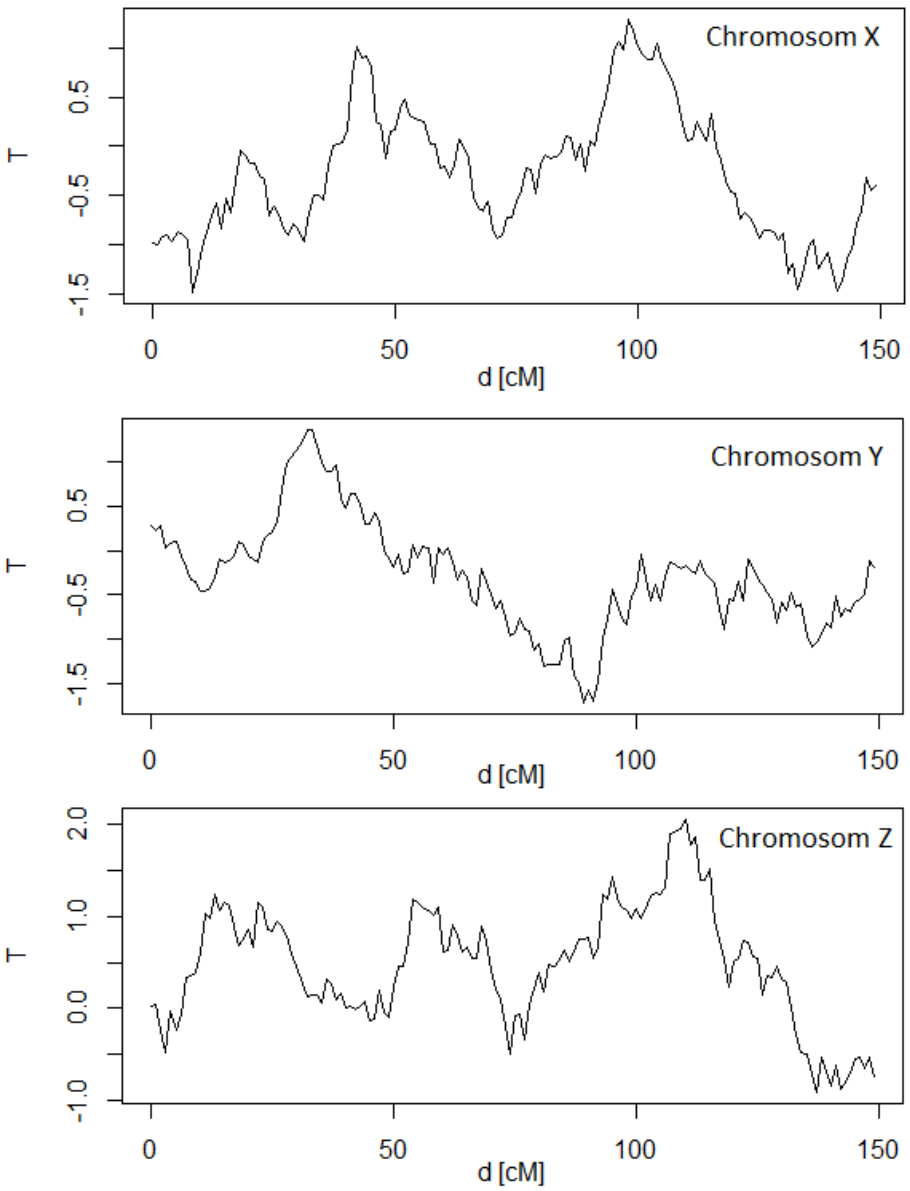
Uzyskane korelacje próbkowe są zbliżone do wartości wyznaczonej teoretycznie, zatem generacja macierzy genotypów przebiegła poprawnie.

Kod źródłowy zadania 1 w języku R:

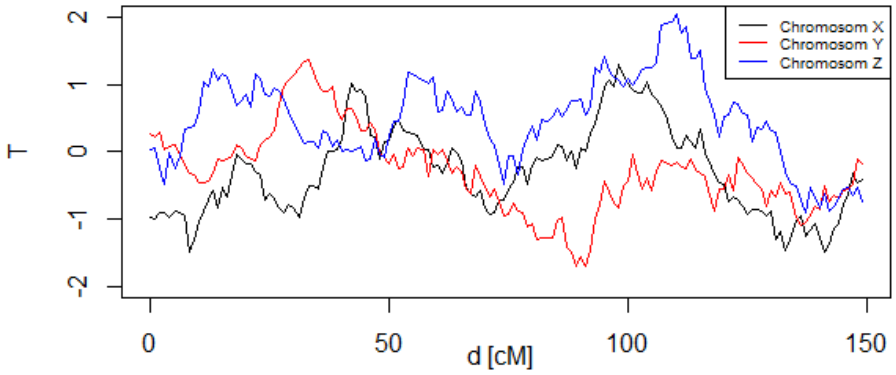
```
# set.seed(124) -> Chromosom Z
# set.seed(122) -> Chromosom Y
set.seed(121) # -> Chromosom X
# ------------------------------------------------
# Autor: M. Bogdan
# Generacja genotypow na jednym chromosomie z krzyzowki
    wstecznej
n <- 500 # liczba osobnikow - wiersze
L <- 150 # dlugosc chromosomu w cM - liczba markerow,
    kolumny;
d <- 1 # odleglosc miedzy sasiednimi markerami w cM
r <- 0.5*(1-exp(-0.02*d)) # p-stwo rekombinacji miedzy
    sasiednimi markerami
P <- rbinom(n,1,0.5) # genotypy w pierwszym markerze
R <- rbinom(n*(L-1),1,r)
R <- matrix(R,nrow=n,ncol=(L-1)) # macierz rekombinacji
    miedzy sasiednimi markerami
X <- cbind2(P,R)
X <- apply(X,1,'cumsum')
X <- t(X)
X <- X%%2 # finalna macierz genotypow
# ------------------------------------------------
# odleglosc miedzy 1. i 5. markerem d = 4
kor_teo = exp(-0.02*4) # 0.9231163

# pierwszy chromosom
kor_expX = cor(X[,1],X[,5]) # 0.9120873

# drugi chromosom
kor_expY = cor(Y[,1],Y[,5]) # 0.9355299

# trzeci chromosom
kor_expZ = cor(Z[,1],Z[,5]) # 0.9199488
```

3 Zadanie drugie

Na samym początku wygenerowano wektor wartości cechy dla $n = 500$ osobników ze standardowego rozkładu normalnego (cecha nie jest zależna od czynników genetycznych). Dla każdego markera (kolumny wygenerowanych macierzy genotypów) dokonano rozbicia próby na dwie grupy w zależności od wartości (0 lub 1) występującej na odpowiedniej pozycji w macierzy genotypów. Następnie dla tak przygotowanego markera wyliczono statystykę testu Studenta. Jej wartość w zależności od odległości od lewego końca chromosomu przedstawiono na wykresach nr 2 oraz 3.



Rysunek 2: Wykresy przedstawiające zależność wartości statystyki Studenta do testowania hipotezy o braku zależności między cechą a genotypem od odległości od lewego końca chromosomów X, Y i Z.



Rysunek 3: Zależność wartości statystyki Studenta do testowania hipotezy o braku zależności między cechą a genotypem od odległości od lewego końca chromosomu.

Gdy rozkład cechy nie odbiega istotnie od normalnego możemy użyć klasycznego testu Studenta (jeżeli rozważamy jedynie dwie wersje genotypu), gdzie hipoteza zerowa mówi nam, iż średnia wartość cechy nie zależy od genotypu markera. Gdy rozkład cechy nie jest normalny możemy zastosować test Wilcoxona lub ewentualnie zamiast wartości cechy rozważać ich rangi.

Kod źródłowy zadania 2 w języku R:

```
set.seed(500)

odleglosc = c(0:149)
cecha = rnorm(500)
StatTX = rep(0,150) # wartosci statystyki Studenta dla
    chromosomu X

for (i in 1:150){
  gr1 = (X[,i] == 0) * cecha
  gr1 = gr1[gr1 != 0]
  gr2 = (X[,i] == 1) * cecha
  gr2 = gr2[gr2 != 0]

  StatTX[i] = t.test(gr1, gr2, var.equal = TRUE)$
      statistic
}

plot(odleglosc, StatTX, type = 'l', xlab = 'd [cM]', ylab
    = 'T')
```

4 Zadanie trzecie

Stosując testy w pojedynczych markerach musimy zmierzyć się z problemem wielokrotnego testowania. Przeprowadzając pojedynczy test na poziomie istotności α nie mamy gwarancji, że utrzymamy ten poziom, wykonując wiele testów.

Aby kontrolować prawdopodobieństwo popełnienia co najmniej jednego błędu pierwszego rodzaju (FWER) stosuje się korekty na wielokrotne testowanie. Najprostszą jest korekta Bonferroniego, w której każdy test wykonujemy na poziomie α/m, gdzie m jest liczbą markerów. Ta korekta okazuje się być dość problematyczna, gdy genotypy markerów są mocno skorelowane i poziom α/m okazuje się być zbyt niski.

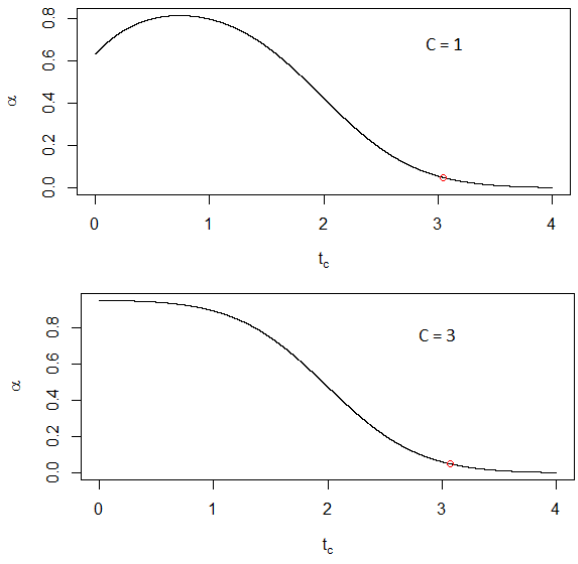
Innym rozwiązaniem jest zastosowanie testów permutacyjnych, które dostosowują wartość krytyczną testu do struktury korelacji między markerami (wartościami statystyk). Wektor cechy permutujemy wielokrotnie, dla każdej permutacji liczymy statystykę testową i wyznaczamy ich maksimum. Za wartość krytyczną uznaje się kwantyl rzędu $1-\alpha$ z rozkładu tak powstałych statystyk.

Gdy rozważaniom poddajemy krzyżówki wsteczne możemy zastosować aproksymację rozkładu statystyk ilorazu wiarygodności na całym chromosomie za pomocą kwadratu procesu Ohrensteina-Uhlenbecka (Feingold, Brown, Siegmund), dzięki czemu wartość krytyczną t_c dla pojedynczego testu policzymy numerycznie z oszacowania

$$\alpha \approx 1 - \exp\left[-2C\{1-\Phi(t_c)\} - 0.04Lt_c\phi(t_c)\nu\left(t_c\sqrt{0.04\Delta}\right)\right],$$

gdzie $\Delta = 1$ oznacza odległość między sąsiednimi markerami (w cM), C jest liczbą chromosomów, L długością chromosomu, a $\nu(t)$ jest zadana wzorem

$$\nu(t) = 2t^{-2}\exp\left\{-2\sum_{n=1}^{\infty} n^{-1}\Phi(-|t|n^{1/2}/2)\right\} \approx \frac{(2/t)(\Phi(t/2)-0.5)}{(t/2)\Phi(t/2)+\phi(t/2)}.$$



Rysunek 4: Zależność wyprowadzona przez Feingolda, Browna i Siegmunda (1993) dla $C = 1$ oraz $C = 3$ z zaznaczoną wartością krytyczną t_c.

Wyestymowane prawdopodobieństwa jednej błędnej detekcji:

Metoda	Macierz X	Macierz Y	Macierz Z	Macierz XYZ
Test Studenta	0.460	0.506	0.472	0.856
Korekta Bonferroniego	0.012	0.008	0.008	0.008
wzór F.,B.,S.	0.044	0.044	0.048	0.112
Test permutacyjny	0.02	0.008	0.012	0.008

Jeżeli każdy test Studenta przeprowadzamy na poziomie istotności 0.05, to oczekiwana liczba detekcji wynosi l. markerów * p. Prawdopodobieństwo co najmniej jednej błędnej detekcji w sytuacji, gdy wielokrotnie wykonano test Studenta jest dużo większe od założonego poziomu istotności i silnie zależy od liczby rozważanych markerów (im więcej markerów bierzemy pod uwagę, tym estymowane prawdopodobieństwo jest większe).

Przy zastosowaniu korekty Bonferroniego frakcja fałszywych odkryć jest mniejsza od $\alpha = 0.05$. Poziom α/m okazuje się być zbyt niski, a tym samym korekta jest nadmiarowa.

Lepszym rozwiązaniem jest zastosowanie wzoru Siegmunda, Browna i Feingolda. Zastosowanym w nim uproszczeniem jest założenie, że odległości między sąsiednimi markerami są jednakowe (co w powyższym przypadku jest prawdą). Dla macierzy X, Y oraz Z otrzymane wyniki są bliskie wartości przyjętego poziomu istotności. Przy złączeniu chromosomów (potrojeniu liczby markerów) wychodzi ono ponad dwa razy większe.

Testy permutacyjne dla macierzy X, Y i Z przeprowadzono w oparciu o 100 permutacji wektora wartości cechy, natomiast dla macierzy połączonej - 1000 permutacji. Proces ten był dość długotrwały (trwał ok. 12h). Warto w tym miejscu podkreślić, że testy permutacyjne mają istotne ograniczenia ilościowe (potrzeba alokacji dużej ilości pamięci). Otrzymane dla danego przypadku wyniki są zbliżone do prawdopodobieństw otrzymanych przy zastosowaniu korekty Bonferroniego.

Kod źródłowy zadania 3 w języku R:

```
# Oszacowanie wartosci krytycznej tc (podpunkt c)
ni <- function(t){
  wn = (2/t)*(pnorm(t/2) -0.5)/((t/2)*pnorm(t/2)+dnorm(t/
      2))
  return (wn)
}
FBS <- function(C,L,delta,tc){
  ff = 1 - exp(-2*C*(1-pnorm(tc)) -0.04*L*tc*dnorm(tc)*ni(
      tc*sqrt(0.04*delta)))
  return (ff)
}
tt = seq(0.001, 4.001, by = 0.0001)
fun = rep(0, length(tt))
for (index in 1:length(tt)){
  fun[index] = FBS(3, 150, 1, tt[index])
}

plot(tt,fun, type = 'l', xlab = expression(t[c]), ylab =
    expression(alpha))
```

```
szukane = 0.05
pozycja = 1
for (kk in 1:length(fun))
{
  if(abs(fun[kk] - szukane) < abs(fun[pozycja] - szukane)
     ){
    pozycja = kk
  }
}
critical_value = tt[pozycja] # tt[30726] = 3.0735 oraz
    3.0435 dla C = 1
lines(critical_value, fun[pozycja], type ='p', col='red')
# ------------------------------------------------------
set.seed(10)

kwantyl=function(alpha, dane){
  dane1=sort(dane)
  return (dane1[floor((1-alpha)*length(dane))])
}

macierz_genotypow = cbind(X,Y,Z) # 3 zlaczone chromosomy
k = 500
powt = 1000
alfa = 0.05

osobniki = length(macierz_genotypow[,1]) # liczba
    osobnikow = 500
markery = length(macierz_genotypow[1,]) # liczba markerow
     = 450
cechy = matrix(rnorm(osobniki*k),osobniki,k)

zliczA = 0 # 428
zliczB = 0 # 4
zliczC = 0 # 56
zliczD = 0 # 4
for (indeks in 1:k) {
  cecha = cechy[,indeks]
  pwar = rep(0,markery)
  stat = rep(0,markery)
  ST = matrix(0, powt, markery)
  for (i in 1:markery){
    gr1 = (macierz_genotypow[,i] == 0) * cecha
    gr1 = gr1[gr1 != 0]
    gr2 = (macierz_genotypow[,i] == 1) * cecha
    gr2 = gr2[gr2 != 0]
    pwar[i] = t.test(gr1, gr2, var.equal = TRUE)$p.value
    stat[i] = t.test(gr1, gr2, var.equal = TRUE)$
        statistic
```

```
    ST[1,i] = stat[i]
    ks[1,i] = abs(mean(gr2)-mean(gr1))
    cem = matrix(sample(cecha), powt-1, osobniki)

    for (jj in 2:powt){
      ce = cem[(jj-1),]
      gru1 = (macierz_genotypow[,i] == 0) * ce
      gru1 = gru1[gru1 != 0]
      gru2 = (macierz_genotypow[,i] == 1) * ce
      gru2 = gru2[gru2 != 0]
      ST[jj,i] = abs(t.test(gru1, gru2, var.equal = TRUE)
          $statistic)
      ks[jj,i] = abs(mean(gr2)-mean(gr1))
    }
  }
  maxD = apply(ST, 2, max)
  critical_value2 = kwantyl(alfa, maxD)
  if (length(pwar[(pwar<alfa) == TRUE]) != 0){
    zliczA = zliczA + 1
  }
  if (length(pwar[(pwar<alfa/markery) == TRUE]) != 0){
    zliczB = zliczB + 1
  }
  if (length(stat[(abs(stat)>critical_value) == TRUE]) !=
      0){
    zliczC = zliczC + 1
  }
  if (length(stat[(abs(stat)>critical_value2) == TRUE]) !
     = 0){
    zliczD = zliczD + 1
  }
}

pA = zliczA/k # 0.856
pB = zliczB/k # 0.008
pC = zliczC/k # 0.112
pD = zliczD/k # 0.008 dla 1000 powtorzen

# wyniki
pA
pB
pC
pD
```

Literatura

[1] M. Bogdan, *Notatki z wykładów Statystyki w zastosowaniach*, Uniwersytet Wrocławski, Wrocław 2016.

[2] K. Dyba, *Notatki z laboratoriów Statystyki w zastosowaniach*, Uniwersytet Wrocławski, Wrocław 2016.

[3] P. Szulc, *Localization og genes*, Mathematica Applicanda Vol. 43(1) 2015, p. 19-35.