

Züchtungslehre – Übung 8

Lösung

04. Dezember 2015

Aufgabe 1

Das Kopplungsungleichgewicht (Linkage Disequilibrium – LD) zwischen Marker und QTL ist eine wichtige Voraussetzung für populationsweite Assoziationsanalysen. Gegeben ist ein Datensatz von 224 Individuen, welche für 8 SNP Marker genotypisiert wurden. Der Datensatz ist in der Datei SNP_LD.txt gespeichert. Der Datensatz kann hier <http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/> unter weitere Materialien runtergeladen werden. Mit diesem Datensatz sollen verschiedene LD-Berechnungen in R durchgeführt werden. Installieren Sie dazu die R-packages „LDheatmap“ und „genetics“.

Manual und Informationen zu den Packages gibt es hier:

<https://cran.r-project.org/web/packages/LDheatmap/LDheatmap.pdf>

<https://cran.r-project.org/web/packages/genetics/genetics.pdf>

Packages installieren:

```
install.packages("LDheatmap")
```

```
install.packages("genetics")
```

Packages laden:

```
library(LDheatmap)
```

```
library(genetics)
```

Lesen den Datensatz in R mit *read.table* ein. Jeder SNP ist mit zwei Allelen im Datensatz enthalten. Erstellen Sie mit dem Befehl „genotype“ ein Genotypenobjekt für jeden SNP: z.B.

```
snp1<-genotype(geno$Snp1_1,geno$Snp1_2)
```

....

Berechnen Sie das paarweise LD (D und r^2) für SNP-Paare mit der Funktion LD:

```
LD(snp1,snp2)$"D"
```

```
LD(snp1,snp2)$"R^2"
```

```
LD(snp1,snp2)
```

Führen Sie die Genotypenobjekte aller SNP in ein data.frame zusammen und berechnen Sie das paarweise LD mit der Funktion LD auf das gesamte data.frame.

Stellen Sie das paarweise LD mittels „LDheatmap“ grafisch dar:

```
LDheatmap(data.frame)
```

```
setwd("/Pfad-zu-den-Daten/")
```

Packages „LDheatmap“ und „genetics“ installieren und laden (siehe oben).

#Daten einlesen

```
genotypes<-read.table("SNP_LD.txt",sep=" ",header=T)
```

#Erstelle Genotypenobjekt für jeden SNP

```
snp1<-genotype(genotypes$Snp1_1,genotypes$Snp1_2)
snp2<-genotype(genotypes$Snp2_1,genotypes$Snp2_2)
snp3<-genotype(genotypes$Snp3_1,genotypes$Snp3_2)
snp4<-genotype(genotypes$Snp4_1,genotypes$Snp4_2)
snp5<-genotype(genotypes$Snp5_1,genotypes$Snp5_2)
snp6<-genotype(genotypes$Snp6_1,genotypes$Snp6_2)
snp7<-genotype(genotypes$Snp7_1,genotypes$Snp7_2)
snp8<-genotype(genotypes$Snp8_1,genotypes$Snp8_2)
```

#LD berechnen für SNP-Paare:

```
> LD(snp1,snp2)
```

Pairwise LD

```
          D          D'          Corr
Estimates: 0.02349079 0.9967644 0.7729406
```

```
          X^2 P-value  N
LD Test: 267.6518    0 224
```

```
> LD(snp1,snp2)$"D"
[1] 0.02349079
```

```
> LD(snp1,snp2)$"R^2"
[1] 0.5974371
```

#Genotypenobjekte aller SNP in ein data.frame zusammenführen:

```
genoall<-(data.frame(snp1,snp2,snp3,snp4,snp5,snp6,snp7,snp8))
```

#Paarweises LD für gesamtes data.frame genoall berechnen

```
> LD(genoall)
> LD(genoall)$"D"
> LD(genoall)$"R^2"
```

#Paarweises LD in einer Heatmap grafisch darstellen:

```
> LDheatmap(genoall)
```

Aufgabe 2

Zehn SNP Marker sollen auf Assoziation mit einem Merkmal getestet werden. 325 Individuen wurden für diese 10 SNP Marker genotypisiert. Die Phänotypen befinden sich in der Datei yvec.txt. Die Datei xmatrix.txt enthält die Designmatrix der SNP-Genotypen und map_markers.txt enthält die Position der Marker im Genom. Die Dateien können hier downgeloaded werden: <http://charlotte-ngs.github.io/LivestockBreedingAndGenomics/>

Lesen Sie alle Dateien mit *read.table* in R ein. Testen Sie den Einfluss eines jeden SNP auf den Phänotyp mittels Single-SNP Regression (*lm*). Welcher SNP weist den grössten Effekt auf? Welcher SNP hat einen signifikanten Einfluss (p-Werte!)?

```
setwd("/Pfad-zu-den-Daten/")
```

#Einlesen der Daten (Phänotypen, Genotypen und Markerkarte mit Positionen der SNP)

```
phenotypes <- read.table("yvec.inp", header=F)
genotypes <- read.table("xmatrix.inp", header=F)
map <- read.table("map_markers.txt", header=F)
```

#Testen des Einflusses des ersten SNP auf den Phänotyp mittels Single-SNP Regression

```
lm(phenotypes[,1]~genotypes[,1])
```

Call:

```
lm(formula = phenotypes[, 1] ~ genotypes[, 1])
```

Coefficients:

```
(Intercept) genotypes[, 1]
  0.1013      0.3826
```

#Erweiterten Modelloutput erhalten wir mit „summary“:

```
summary(lm(phenotypes[,1]~genotypes[,1]))
```

Call:

```
lm(formula = phenotypes[, 1] ~ genotypes[, 1])
```

Residuals:

```
   Min     1Q  Median     3Q    Max
-1.04889 -0.31149 -0.00549  0.30911  1.41251
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.10128   0.03897   2.599 0.00979 **
genotypes[, 1] 0.38261   0.03338  11.462 < 2e-16 ***
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4509 on 323 degrees of freedom
Multiple R-squared: 0.2891, Adjusted R-squared: 0.2869
F-statistic: 131.4 on 1 and 323 DF, p-value: < 2.2e-16

#Nur den Effekt des SNP ausgeben (Regressionskoeffizient)

```
lm(phenotypes[,1]~genotypes[,1])$coeff[2]
genotypes[, 1]
0.382608
```

#Ausgabe des genauen P-Wertes (Wie hoch ist der P-Wert genau?)

Dazu kann eine Anova durchgeführt werden. „anova“ steht für Analysis of variance (wird oft für Modellvergleiche angewandt):

```
anova(lm(phenotypes[,1]~genotypes[,1]))
Analysis of Variance Table
```

```
Response: phenotypes[, 1]
          Df Sum Sq Mean Sq F value Pr(>F)
genotypes[, 1] 1 26.708 26.7076 131.37 < 2.2e-16 ***
Residuals    323 65.665  0.2033
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(phenotypes[,1]~genotypes[,1]))$P[1]
[1] 9.492982e-26
```

#Testen aller SNP im Loop und speichern der Ergebnisse in einem data.frame

Zuerst wird die Dimension des data.frames definiert:

```
results <- data.frame(rep(NA,10),rep(NA,10))
names(results) <- c(„Effekt“, „p-Wert“)
```

Loop über die 10 SNP

```
for (i in 1:10){
  results[i,1] <- lm(phenotypes[,1]~genotypes[,i])$coeff[2]
  results[i,2] <- anova(lm(phenotypes[,1]~genotypes[,i]))$P[1]
}
```

results

	Effekt	p-Wert
1	0.3826080	9.492982e-26
2	0.4535504	6.820006e-13
3	0.4535504	6.820006e-13
4	0.4972216	7.047841e-43
5	0.4465850	8.634177e-30
6	0.4926920	1.615060e-37
7	0.2221599	1.632815e-04
8	-0.4972216	7.047841e-43
9	0.3527940	3.354479e-13
10	-0.4972216	7.047841e-43

#Visualisierung der Ergebnisse: Plotten der P-Werte gegen der Position der SNP im Genom

```
plot(map[,2],-log10(results[,2]), xlab="snp position", ylab="-log10(p-value)")
```