Modul Statistische Aspekte der Analyse molekularbiologischer und genetischer Daten

R-Blatt 4: Visualisierung statistischer Konzepte

Janne Pott

WS 2021/22

In dieser Übung wird R genutzt, um verschiedene statistische Konzepte zu visualisieren.

Verwandtschaft

Beispiele

Zur paarweisen Schätzung von Verwandtschaften haben Sie in der Vorlesung den Kinship-Schätzer kennen gelernt:

$$\hat{k}_{i,j} = \frac{1}{M} \sum_{m=1}^{M} \frac{(g_{m,i} - 2 * p_{m,B})(g_{m,j} - 2 * p_{m,B})}{4 * p_{m,B} * p_{m,A}}$$

mit

- M als Anzahl der betrachteten biallelischen SNPs (Allel A und B)
- $p_{m,B}$ als Allelfrequenz des SNPs m bezüglich Allel B
- $g_{m,i}$ als Genotyp des SNPs m von Person i bezüglich Allel B

In diesem Beispiel verwenden wir den Datensatz *verwandtschaft.RData*.

```
loaded<-load("data/verwandtschaft.RData")
loaded
```

```
## [1] "allelfreq" "genotypes"
```

```
dim(genotypes)
```

```
## [1] 30000    10
```

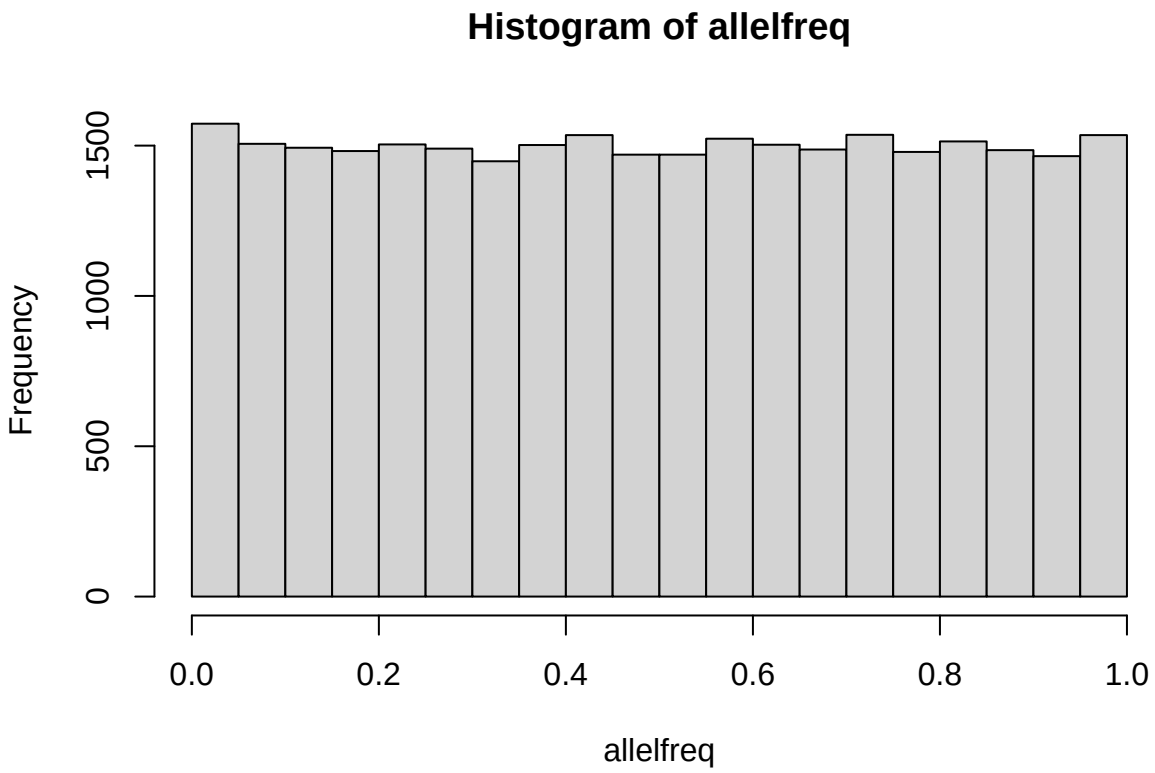
```
class(genotypes)
```

```
## [1] "matrix" "array"
```

```
length(allelfreq)
```

```
## [1] 30000
```

```
hist(allelfreq)
```



Im Datensatz sind zwei Parameter, einmal *genotypes*, eine Genotyp-Matrix von 10 Personen mit 30,000 SNPs, und *allelfreq*, ein Vektor der Allelfrequenzen pro SNP aus *genotypes* bezüglich Allel B.

Im Histogramm kann man erkennen, dass die Allelfrequenzen in etwa gleichverteilt sind. Das heißt, dass die *rare* SNPs (Allelfrequenz<5%) gleich-häufig sind wie die *common* SNPs.

Um den Kinship-Schätzer zu bestimmen, kann man zwei Schleifen für Person i und j nutzen:

```
p2<-2*allelfreq
p4<-4*allelfreq*(1-allelfreq)

K<-matrix(NA,10,10)
for(i in 1:10){
  # i=1
  for(j in i:10){
    # j=2
    g_i<-genotypes[,i]-p2
    g_j<-genotypes[,j]-p2
    x<-sum(g_i*g_j/p4)
    k_ij<-x/30000
```

```
    K[i,j]<-k_ij
    K[j,i]<-k_ij
  }

}

knitr::kable(K,digits = 3, caption = "Verwandschaftsmatrix mittels Schleife")
```

Table 1: Verwandschaftsmatrix mittels Schleife

0.496	-0.002	0.001	-0.002	0.243	0.243	0.245	0.248	0.248	0.243
-0.002	0.501	0.000	0.001	0.244	0.245	-0.002	-0.002	0.001	-0.003
0.001	0.000	0.500	-0.003	-0.001	-0.002	0.247	0.252	-0.001	-0.002
-0.002	0.001	-0.003	0.500	0.000	0.001	-0.001	-0.003	0.249	0.250
0.243	0.244	-0.001	0.000	0.488	0.238	0.120	0.119	0.130	0.119
0.243	0.245	-0.002	0.001	0.238	0.490	0.121	0.120	0.121	0.122
0.245	-0.002	0.247	-0.001	0.120	0.121	0.492	0.244	0.119	0.120
0.248	-0.002	0.252	-0.003	0.119	0.120	0.244	0.498	0.122	0.121
0.248	0.001	-0.001	0.249	0.130	0.121	0.119	0.122	0.497	0.245
0.243	-0.003	-0.002	0.250	0.119	0.122	0.120	0.121	0.245	0.490

Aufgaben

a) Die Verwandtschaftsmatrix kann auch mittels Matrix-Operation bestimmt werden. Definieren Sie dazu eine Hilfsmatrix h, die auf die Allelfrequenzen und Anzahl der SNPs adjustiert ist, und bilden Sie das Matrixprodukt $H = h^T * h$. Stimmt dieses Produkt mit K überein?

$$h_{m,i} = \frac{(g_{m,i} - 2 * p_{m,B})}{\sqrt{M * 4 * p_{m,B} * p_{m,A}}}$$

b) Warum gilt:

$$\hat{k}_{i,j} \approx 0.5$$

c) Wie viele paarweise Verwandtschaften (von Grad 1,2, ... , unverwandt) beobachten Sie?

d) Welche Familienstruktur könnte die beobachteten Verwandtschaftsbeziehungen erklären?

XY-Plot

Beispiele

Bei der Genotypisierung mittels SNP-Array kann meistens das genetische Geschlecht bestimmt werden. In wenigen Ausfällen ist die Intensität der X- bzw. Y-SNPs zu verrauscht um eine Aussage zu treffen. Diese werden dann als *unknown* klassifiziert. Um hier zu entscheiden, ob ein Sample gefiltert werden muss, benutzt man XY-Plots.

```
loaded<-load("data/XYPlots.RData")
loaded
```

```
## [1] "daten"   "samples"
```

```
dim(samples);colnames(samples)
```

```
## [1] 300   3
```

```
## [1] "sampleID"      "sex_datenbank" "sex_computed"
```

```
rownames(samples)[c(1:10,291:300)]
```

```
##  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "291" "292"
## [13] "293" "294" "295" "296" "297" "298" "299" "300"
```

```
dim(daten);colnames(daten)[c(1,200,201,300)]
```

```
## [1] 900 300
```

```
## [1] "X:1"   "X:200" "Y:1"   "Y:100"
```

```
rownames(daten)[1:3]
```

```
## [1] "1:genotype" "1:intA"     "1:intB"
```

Für dieses Beispiel liegen Daten von 300 SNPs und 300 Samples vor. Zu den Samples (IDs 1-300) ist das Geschlecht mitangegeben, einmal aus der Datenbank (*sex_datenbank*) und einmal wie es beim Calling bestimmt wurde (*sex_computed*). Die SNPs haben die IDs mit Chromosomenangabe, d.h. die X-chromosomalen SNPs haben die IDs *X:1 - X:200*, die Y-chromosomalen *Y:1 - Y:100*. Pro SNP und Sample liegen die Intensitäten der Allele A und B und der wahrscheinlichste Genotyp vor (0=AA, 1=AB, 2=BB, -1=NA).

Die X-Heterozygosität ist ein weiterer Marker der Qualität: Männer haben nur ein X, daher sollten sie eine Heterozygosität von 0 haben. Für Frauen erwartet man 0.25 (Erwartungswert der Heterozygoten bei eine Beta-Verteilung mit $\alpha = \beta = 0.5$).

Herleitung des Erwartungswert mit beta-verteilter Allelfrequenz x:

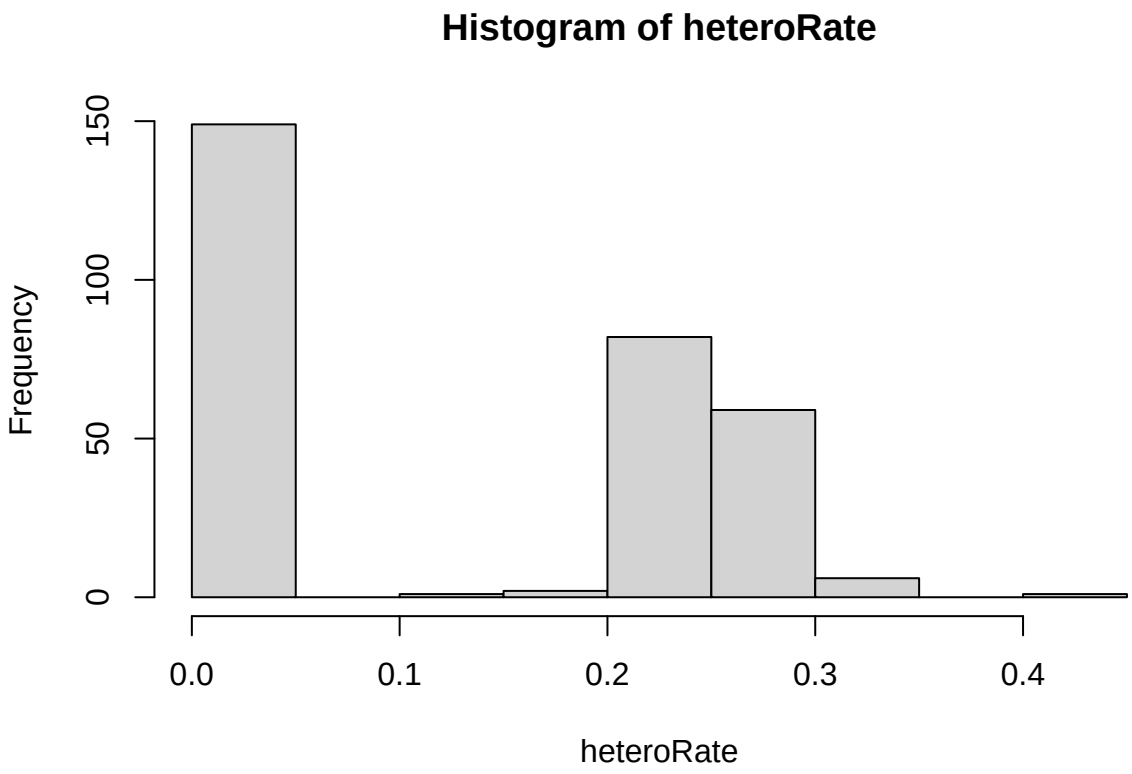
$$
\begin{aligned}
E(X = AB) &= \int 2 * x * (1-x) * \frac{x^{\alpha-1} * (1-x)^{\beta-1}}{B(\alpha,\beta)} dx \\
&= \int \frac{2}{B(\alpha,\beta)} x^{\alpha} * (1-x)^{\beta} dx \\
&= \frac{2}{B(\alpha,\beta)} * \int x^{\alpha} * (1-x)^{\beta} dx \\
&= \frac{2 * B(\alpha+1, \beta+1)}{B(\alpha,\beta)} \\
&= 2 * 0.3926991/3.141593 = 0.25
\end{aligned}
$$

```
filt<-grepl("genotype",rownames(daten))
geno<-daten[filt,]
intent<-daten[!filt,]

snpDataX<-geno[,1:200]
dim(snpDataX)
```

```
## [1] 300 200
```

```
snpDataX<-t(snpDataX)
countGenos<-function(x) {return(as.numeric(table(factor(x,levels=-1:2))))}
nrGenos<-apply(snpDataX,2,countGenos)
nrGenos<-t(nrGenos)
colnames(nrGenos)<-c("miss","AA","AB","BB")
heteroRate<-apply(nrGenos,1,function(x) x[3]/sum(x[2:4]))
hist(heteroRate)
```



Aufgaben

a) Für den XY-Plot brauchen wir die Gesamtintensitäten. Bilden Sie daher zuerst den Mittelwert der Intensität für Allel A und B pro SNP und bilden Sie dann die jeweilige mittlere Intensität aller X-SNPs und aller Y-SNPs pro Sample!

b) Sie sollten nun ein Objekt mit den Variablen *ID*, *X-Intensität*, *Y-Intensität, X-Heterozygosität*, *sex_datenbank* und *sex_computed* pro Sample haben. Erzeugen Sie nun folgende drei Plots und markieren Sie in diesen Plots Ausreißer (widersprüchliche Geschlechtsangabe, zu hohe/niedrige Intensitäten, auffällige Heterozygosität):

   i. X-Intensität – Y-Intensität

   ii. X-Intensität – X-Heterozygosität

   iii. Y-Intensität – X-Heterozygosität

Principal Component Analysis (PCA)

In dieser Aufgabe sollen die Eigenvektoren von genetischen Daten erzeugt werden. Dafür wird ein Teil der Daten von 1000 Genomes (1KG, Phase 1, release 3) verwendet. Zusätzlich zu R wird hier PLINK verwendet, da dieses Programm effizienter große Datenmengen verarbeiten kann.

Diese Aufgabe ist als Tutorial aufgebaut: jeder Schritt wird angefangen und soll von Ihnen vervollständigt werden.

Als erstes soll der Pfad zu PLINK definiert werden. Als kleiner Test wird Plink einmal aufgerufen.

- Falls 127 zurückggeben wird, hat R die Plink .exe nicht gefunden - bitte Pfad prüfen!

```
plink_call<-"C:/Programme/plink2/plink2"

# test if plink can start
system(plink_call)
```

```
## [1] 1
```

```
# print plink start (by default printed in the render console, not the pdf or html output)
cat(system(plink_call, intern = TRUE), sep = '\n')
```

```
## Warning in system(plink_call, intern = TRUE): Ausführung von Kommando 'C:/
## Programme/plink2/plink2' ergab Status 1

## PLINK v2.00a3 64-bit (11 Oct 2021)             www.cog-genomics.org/plink/2.0/
## (C) 2005-2021 Shaun Purcell, Christopher Chang   GNU General Public License v3
##
##   plink2 <input flag(s)...> [command flag(s)...] [other flag(s)...]
##   plink2 --help [flag name(s)...]
##
## Commands include --rm-dup list, --make-bpgen, --export, --freq, --geno-counts,
## --sample-counts, --missing, --hardy, --het, --fst, --indep-pairwise, --ld,
## --sample-diff, --make-king, --king-cutoff, --pmerge, --pgen-diff,
## --write-samples, --write-snplist, --make-grm-list, --pca, --glm, --adjust-file,
## --score, --variant-score, --genotyping-rate, --pgen-info, --validate, and
## --zst-decompress.
##
## "plink2 --help | more" describes all functions.
```

Datenvorbereitung - SNPs filtern

Überprüfen Sie in R, ob alle SNPs von *mySNPs.txt* in 1KG sind. Hierfür sollte man am besten das *1KG_PCA.bim* File verwenden (tab-delimited). Recherchieren Sie, was in den einzelnen Spalten des .bim Files steht. Nutzen sie zum Einlesen der Befehl *fread()* aus dem Paket „*data.table*“. Filtern Sie nach den SNPs in der Schnittmenge und erstellen Sie ein gefiltertes Text-File mySNPs_filtered.txt!

Hinweis: Es sollten am Ende 206,233 SNPs sein!

Spalte	Information
V1	
V2	
V3	
V4	
V5	
V6	

```
myTab<-read.table("data/mySnps.txt")
dim(myTab)
```

```
## [1] 224458      1
```

```
rslist<-fread("data/1KG_PCA.bim",sep="\t",stringsAsFactors=F)
dim(rslist)
```

```
## [1] 498586      6
```

```
head(rslist)
```

```
##    V1         V2 V3      V4 V5 V6
## 1:  1  rs3094315  0  752566  G  A
## 2:  1 rs12184325  0  754105  T  C
## 3:  1  rs3131969  0  754182  A  G
## 4:  1 rs12562034  0  768448  A  G
## 5:  1 rs11240777  0  798959  A  G
## 6:  1 rs11579015  0 1036959  C  T
```

```
# to continue ...
```

Datenvorbereitung - Samples filtern

Erstellen Sie eine Sample Liste mit Individuen aus Asien, Afrika und Europa! Nutzen Sie hierfür das **1KG_PCA.fam** File (space-delimited). Wir wollen eine möglichst große Menge an Samples, aber jeder Herkunft sollte gleich oft vorhanden sein! Ziehen Sie zufällig aus der jeweiligen Teilmenge und speichern Sie ihre Liste als **mySamples.txt** ab!

Hinweis: Es sollten am Ende 3*246 Individuen sein!

Spalte	Information
V1	
V2	
V3	
V4	
V5	
V6	

```
fam.data<-read.table("data/1KG_PCA.fam",stringsAsFactors=F,sep=" ")
dim(fam.data)
```

```
## [1] 1092    6
```

```
head(fam.data)
```

```
##        V1          V2 V3 V4 V5 V6
## 1 HG00096 EUR_HG00096  0  0  1 -9
## 2 HG00097 EUR_HG00097  0  0  2 -9
## 3 HG00099 EUR_HG00099  0  0  2 -9
## 4 HG00100 EUR_HG00100  0  0  2 -9
## 5 HG00101 EUR_HG00101  0  0  1 -9
## 6 HG00102 EUR_HG00102  0  0  2 -9
```

```
ethno<-substr(fam.data$V2,1,3)
table(ethno)
```

```
## ethno
## AFR AMR ASN EUR
## 246 181 286 379
```

```
# to continue ...
```

Datenvorbereitung - SNPs prunen

Jetzt prunen Sie die SNPs mit PLINK, d.h. Sie prüfen, welche SNPs in hohem LD miteinander sind. Folgende Parameter sollten Sie setzen: a. Input:–bfile 1KG_PCA b. SNPs einschränken: –extract mySNPs_filtered.txt c. Samples einschränken: –keep mySamples.txt d. LD-Pruning-Parameter festlegen: –indep-pairwise 50 5 0.2 e. Output: –out pruned_filter

Was bedeuten die drei Zahlen hinter dem –indep-pairwise Befehl?

Hinweis: Es sollten am Ende 117,351 SNPs sein.

```
call1<-paste(plink_call,
             "--bfile data/1KG_PCA",
             "--extract PCA/mySnps_filtered.txt",
             "--keep PCA/mySamples.txt",
             "--indep-pairwise 50 5 0.2",
             "--out PCA/pruning_filter")
system(call1)
```

Datenvorbereitung - Datensatz erstellen

Erstellen Sie jetzt mit PLINK ein neues .bed-File, dass nur noch die geprunten SNPs und die gewünschten Samples (aus 2) enthält (–bfile, –extract, –keep, und –make-bed).

```
call2<-paste(plink_call,
             "--bfile data/1KG_PCA",
             "--extract PCA/pruning_filter.prune.in",
             "--keep PCA/mySamples.txt",
             "--make-bed --out PCA/pruned_data")
system(call2)
```

PCA berechnen

Jetzt kann mit den neuen Files die PCA ausgerechnet werden (–bfile, –pca, –out):

```
call3<-paste(plink_call,
             "--bfile PCA/pruned_data",
             "--pca --out PCA/pca_out")
system(call3)
```

PCA auswerten

Laden Sie beide Outputs der PCA in R ein! Wie sind die Daten aufgebaut?

Erstellen Sie einen Plot der ersten beiden Vektoren mit Ethnien-Färbung! Was kann man daraus schließen?

Berechnen Sie den Anteil der erklärten Varianz a. durch den ersten Eigenvektor und b. durch die ersten beiden Eigenvektoren!

Was würden Sie erwarten, wenn alle 4 Ethnien in die Analyse eingeflossen wären? Wo würden Sie die Amerikaner einordnen? Rechnen Sie das nach!

```
pca2values<-read.table("PCA/pca_out.eigenval")$V1
pca2vector<-read.table("PCA/pca_out.eigenvec",stringsAsFactors=F,sep="\t")

# to continue ...
```