



Systems Genetics Exercise 1: Introduction

Thursday, October 21

Matthias Heinig, Corinna Losert, Katharina Schmid

1 General

Exercises belong to two different categories:

- Pen and paper exercise marked with the symbol 
- Programming exercise marked with the symbol 

For questions regarding this exercise, feel free to contact corinna.losert@helmholtz-muenchen.de or katharina.schmid@helmholtz-muenchen.de.

2 Requirements

1. R packages

- VariantAnnotation (install with: **BiocManager::install("VariantAnnotation")**)
- biomaRt (install with: **BiocManager::install("biomaRt")**)
- Gviz (install with: **BiocManager::install("Gviz")**)
- optional but really useful to create pdf-reports: markdown & knitr (needs a valid TeX installation)

2. Data

- filtered 1000 genomes genotypes
vcf file (*e-geuv-1_filtered.vcf.bgz* and *e-geuv-1_filtered.vcf.bgz.tbi*, 1.4 MB / 114 KB)

In case you have problems installing R or any package, have a look at the instructions on moodle (called "installation.pdf") or use google colab instead (you can copy the notebook from this [template](#))

3 Quick R refresher

Several of the following exercises will be in R, so let's make sure you all know the basic R commands. For help, have a look at the large collections of cheatsheets from Rstudio <https://www.rstudio.com/resources/cheatsheets/>, such as the one with base R commands <http://github.com/rstudio/cheatsheets/raw/master/base-r.pdf>.

Try to solve these short exercises to make sure you now basic R commands:

- Load the 'VariantAnnotation' package which will also be used later on in the script
- Get a list of the functions within the package
- Check out the documentation for the function "readVcf"
- Get your current working directory

- Assign the sum of 2,3 and 4 to variable x
- Make a character vector of the gene names PAX6, ZIC2, OCT4 and SOX2 and a second numeric count vector of the same length containing randomly sampled numbers between 1 and 10 (set the seed 42)
- Subset the gene - vector using [] notation, and get the 2nd and 4th element
- Generate a dataframe out of the two generated vectors
- Select the genes of the generated dataframe for which the corresponding count value is greater than 5
- Make a boxplot of the distribution of the generated count values in the dataframe
- Write a function that takes a gene name and a dataframe as input with one column named "genes" and searches whether this gene name occurs in the column. It returns TRUE in case the genename occurs in the dataset and FALSE in case it doesn't. Test the function with the above generated dataset.

```
# Load the 'VariantAnnotation' library which will also be used later on in the script
```

```
library(VariantAnnotation)
```

```
# Get a list of the functions within the package  
# (Here showing only the first 6 with head)
```

```
head(ls("package:VariantAnnotation"))
```

```
## [1] "AllVariants" "alt"          "alt<-"        "altDepth"     "altDepth<-"  
## [6] "altFraction"
```

```
# Check out the documentation for the function "readVcf"
```

```
?readVcf
```

```
# Get your current working directory
```

```
getwd()
```

```
## [1] "/Users/katharina.schmid/Documents/Lehre/SystemGenetics/system_genetics_lecture"
```

```
# Assign the sum of 2,3 and 4 to variable x
```

```
x <- 2+3+4
```

```
# Make a character vector of the gene names PAX6,ZIC2,OCT4 and SOX2 and a  
# second numeric count vector of the same length containing randomly sampled  
# numbers between 1 and 10 (set the seed 42)
```

```
genes <- c('PAX6','ZIC2','OCT4','SOX2')  
set.seed(42)  
counts <- sample(1:10, 4)
```

```
# Subset the gene - vector using [] notation, and get the 2nd and 4th elements
```

```
genes[c(2,4)]
```

```
## [1] "ZIC2" "SOX2"
```

```
# Generate a dataframe out of the two generated vectors
```

```
data <- data.frame(genes, counts)
```

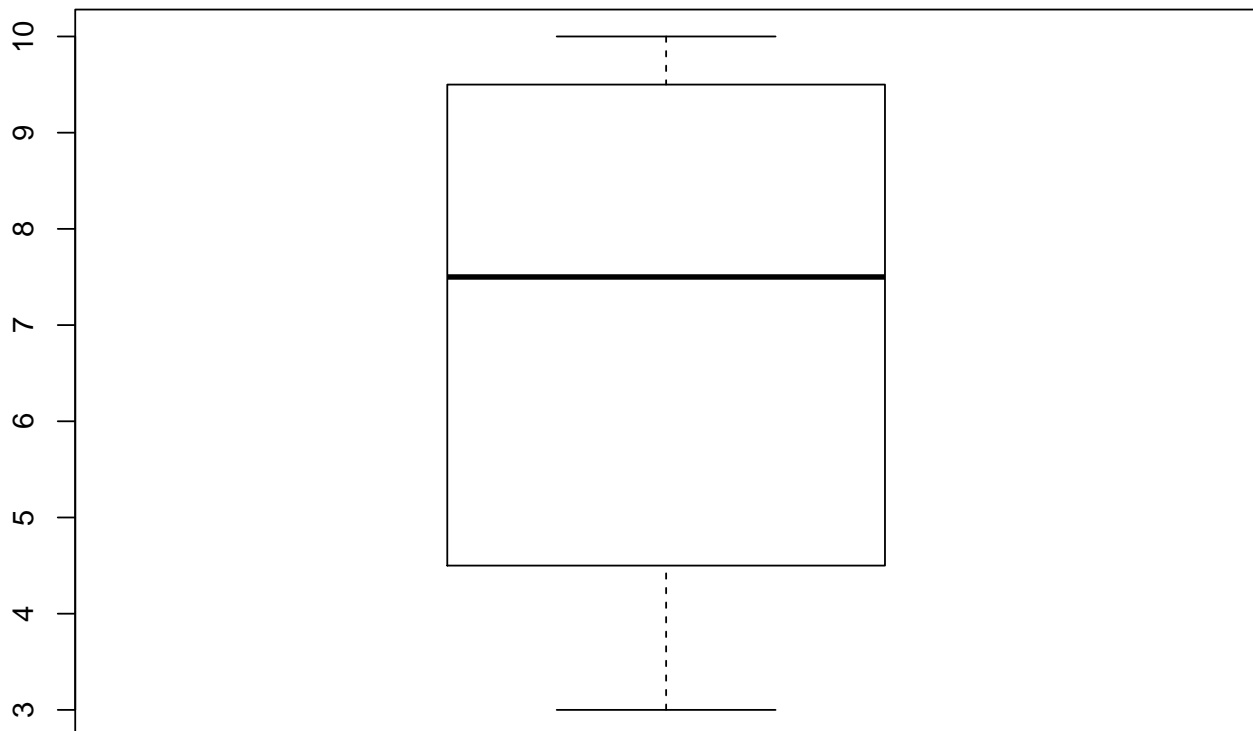
```
# Select the genes of the generated dataframe for which the corresponding  
# count value is greater than 5
```

```
data[data$counts>5,]
```

```
##      genes counts  
## 1  PAX6       10  
## 2  ZIC2        9  
## 4  SOX2        6
```

```
# Make a boxplot of the distribution of the generated count values in the  
# dataframe
```

```
boxplot(data$counts)
```



```
# Write a function that takes a gene name and a dataframe as input with one  
# column named "genes" and searches whether this gene name occurs in the column.  
# It returns TRUE in case the genename occurs in the dataset and FALSE in  
# case it doesn't. Test the function with the above generated dataset.
```

```
gene_check <- function(gene_name , data){  
  
  return(gene_name %in% data$genes)  
}
```

```
gene_check('ABC', data)
```

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

```
gene_check('SOX2', data)
```

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

In case you need further R-practice you can also take a look at this short R-Introduction and exercises: <https://compgenomr.github.io/book/exercises.html>

4 Basic genetic vocabulary

Please explain shortly the following genetic terms:

- Central Dogma of Molecular Biology
- gene
- allele
- genotype
- heterozygous
- phenotype
- SNP

```
# Central Dogma of Molecular Biology = explaining the general flow of genetic
#       information: DNA gets transcribed to RNA which gets translated into proteins
# gene = historically a specific location in the genome; today: a sequence of
#       nucleotides on the DNA that gets transcribed into RNA
#       (and in many cases translated into proteins afterwards)
# allele = a particular version of a gene (e.g. A vs a), for example the "ABO"
#       blood types are caused by a gene with 8 common alleles
# genotype = combination of alleles of a gene (e.g. AA,Aa,aa)
# heterozygous = a genotype with two different alleles (=Aa), in contrast
#       to homozygous (= AA or aa)
# phenotype = the set of observable traits of an organisms, including a wide range
#       of characteristics from height and hair color up to disease status
# SNP = single nucleotide polymorphism, specific case of a genetic variant
#       where only one base is changed; SNPs can have an effect on the phenotype
#       (e.g. cause a disease), but not have to
```

5 VCF (Variant Call format) files

5.1 VCF Format

In the following, you can find the first few lines of a VCF file.

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA000001 NA000002 NA000003
20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3 0/0:41:3
20 1110696 rs6040355 A G,T 67 PASS NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2 2/2:35:4
20 1230237 . T . 47 PASS NS=3;DP=13;AA=T GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20 1234567 microsat1 GTCT G,GTACT 50 PASS NS=3;DP=9;AA=G GT:GQ:DP 0/1:35:4 0/2:17:2 1/1:40:3
```

5.1.1 What is saved in a VCF file?

```
# VCF = Variant Call Format
# File format for saving genetic variants, each row representing one variant and
# columns the samples (starting after the annotation columns)
```

5.1.2 Which are the eight mandatory columns in the header line? Explain their meaning and specify their data format!

```
# CHROM: chromosome identifier (reference genome) - String
# POS: reference position - Integer
# ID: unique identifiers (e.g. rs numbers for SNPs) - String
# REF: reference base; "wild type" - String
# ALT: alternate non-reference allele
# QUAL: quality score,  $-10\log_{10} p([no]variant)$  - Numeric
# FILTER: Filter information - String
# INFO: additional information - String
```

5.1.3 What are the genotypes of samples NA000001, NA000002 and NA000003 for the variant rs6054257 (write down nucleotides)?

```
# Alleles are encoded with 0-1, 0 represents the reference allele and 1 the
# alternative allele

# This leads to the following genotypes:
# NA000001: G G
# NA000002: A G
# NA000003: A A
```

5.2 VCF in R

Read-in the vcf file using the package **VariantAnnotation** and answer the following questions:

- Get the number of samples and variants.
- Get the first 5 SNPs from the first 3 samples.
- Get the reference and alternative alleles for these first 5 SNPs.
- Get the genotypes of samples HG00351, HG00353 and HG00355 for the variant rs17042098
- Get the frequencies of the genotypes for SNP rs17042098
- Convert the genotypes 0/0, 0/1, 1/1 for SNP rs17042098 to 0, 1, 2.

For help, check the Bioconductor documentation:

<http://bioconductor.org/packages/release/bioc/html/VariantAnnotation.html>

```
library(VariantAnnotation)

vcf<-readVcf("data/e-geuv-1_filtered.vcf.bgz")

# get the number of samples and variants
dim(vcf)

## [1] 14113    271

# get the first 5 SNPs from the first 3 samples
geno(vcf)$GT[1:5,1:3]

##           HG00096 HG00097 HG00099
## rs114635344 "0/0"    "0/0"    "0/0"
## rs115807277 "0/0"    "0/0"    "0/0"
## rs72643476  "0/1"    "0/1"    "0/1"
## rs2250799   "1/1"    "0/1"    "1/1"
## rs2477692   "0/1"    "0/1"    "0/1"

# get the reference and alternative alleles for these first 5 SNPs
ref(vcf)[1:5]

##      A DNAStringSet instance of length 5
##      width seq
## [1]      1 G
## [2]      1 A
## [3]      1 G
## [4]      1 G
## [5]      1 C

alt(vcf)[1:5]

## DNAStringSetList of length 5
## [[1]] A
## [[2]] G
## [[3]] A
## [[4]] A
## [[5]] T

# get the genotypes for rs17042098 and samples HG00351, HG00353, HG00355
geno(vcf)$GT["rs17042098",c("HG00351", "HG00353", "HG00355")]
```

```
## HG00351 HG00353 HG00355
## "0/0" "0/1" "0/0"

# get the frequencies of the genotypes for SNP rs17042098 (summarize)
table(geno(vcf)$GT["rs17042098",])

##
## 0/0 0/1 1/1
## 196 68 7

# convert genotypes for SNP rs17042098 to numeric
head(as(genotypeToSnpMatrix(vcf["rs17042098",])$genotype, "numeric"))

##          rs17042098
## HG00096           0
## HG00097           0
## HG00099           0
## HG00100           0
## HG00101           0
## HG00102           0
```

5.3 Genomic Ranges in R

GRanges objects are representations of genomic regions in R, consisting of a chromosome (called seqnames), a start position in base pairs (bp) and a width in bp. Additional information can be added as metadata columns, better describing the regions. For more details, see:

<https://bioconductor.org/packages/release/bioc/vignettes/GenomicRanges/inst/doc/GenomicRangesIntroduction.html>

Get all SNPs in the region of chromosome 4, 2 000 000 - 3 000 000 bp.

Hints:

- you can extract a *GRanges* object containing SNP annotations from the vcf file using **rowRanges()**
- create a *GRanges* object of the region of interest and calculate overlaps with **findOverlaps()**

```
# define the search region as a GRanges object
search.range<-GRanges(seqnames = "4",
                      ranges = IRanges(start = 2000000, width = 1000000))

# get GRanges object with all SNPs annotated in the vcf file
snp.regions<-rowRanges(vcf)

# find overlaps between both GRanges objects and subset the vcf GRanges object
ov<-findOverlaps(snp.regions,search.range)
snp.regions[ov@from]
```

```
## GRanges object with 2 ranges and 5 metadata columns:
##          seqnames   ranges strand | paramRangeID      REF
##          <Rle> <IRanges> <Rle> | <factor> <DNAStringSet>
##   rs3135157         4   2087202   * |      <NA>          A
##   rs3021145         4   2950899   * |      <NA>          T
##                                ALT     QUAL     FILTER
##                                <DNAStringSetList> <numeric> <character>
##   rs3135157                                G         100      PASS
```

```
## rs3021145          C      100      PASS
## -----
## seqinfo: 22 sequences from b37 genome
```

6 Biomart annotation

biomaRt is a package to retrieve gene annotations from Biomart in R.

Use this to look up the position (*chromosome_name*, *start_position*, *end_position*) and HGNC gene symbols for the genes with Ensembl gene ids (*ensembl_gene_id*) ENSG00000196620, ENSG00000109787, ENSG00000241163 and ENSG00000000938.

Hints:

- get ensembl GRCh37 annotations:
`useEnsembl(biomart="ensembl",dataset="hsapiens_gene_ensembl",GRCh=37)`
- select desired genes with `getBM()`

```
library(biomaRt)

# get the basic ensembl annotations based on GRCh37
ensembl = useEnsembl(biomart="ensembl", dataset="hsapiens_gene_ensembl", GRCh=37)

# define genes of interest with ensembl ids
genes <- c("ENSG00000196620", "ENSG00000109787",
           "ENSG00000241163", "ENSG00000000938")

# filter for desired gene ids
anno <- getBM(attributes=c('ensembl_gene_id','chromosome_name',
                           'start_position','end_position','hgnc_symbol'),
              filters = 'ensembl_gene_id', values=genes, mart = ensembl)

anno
```

	ensembl_gene_id	chromosome_name	start_position	end_position	hgnc_symbol
## 1	ENSG00000000938	1	27938575	27961788	FGR
## 2	ENSG00000109787	4	38665817	38702663	KLF3
## 3	ENSG00000196620	4	69512348	69536346	UGT2B15
## 4	ENSG00000241163	3	72084451	72291716	LINC00877

7 Visualization of genomic data

Visualize the surrounding of rs17042098 (+/- 500 000 bp) on the genome using the package Gviz. Possible tracks that you could use are: IdeogramTrack to depict the whole chromosome, GenomeAxisTrack to add bp axis and BiomartGeneRegionTrack to add gene annotations.

Use the user guide for more information:

<https://bioconductor.org/packages/release/bioc/vignettes/Gviz/inst/doc/Gviz.html>

```
library(Gviz)

# define the visualized region around rs17042098
```



```

startRegion <- rowRanges(vcf)["rs17042098"]@ranges@start-5e+05
endRegion <- rowRanges(vcf)["rs17042098"]@ranges@end+5e+05

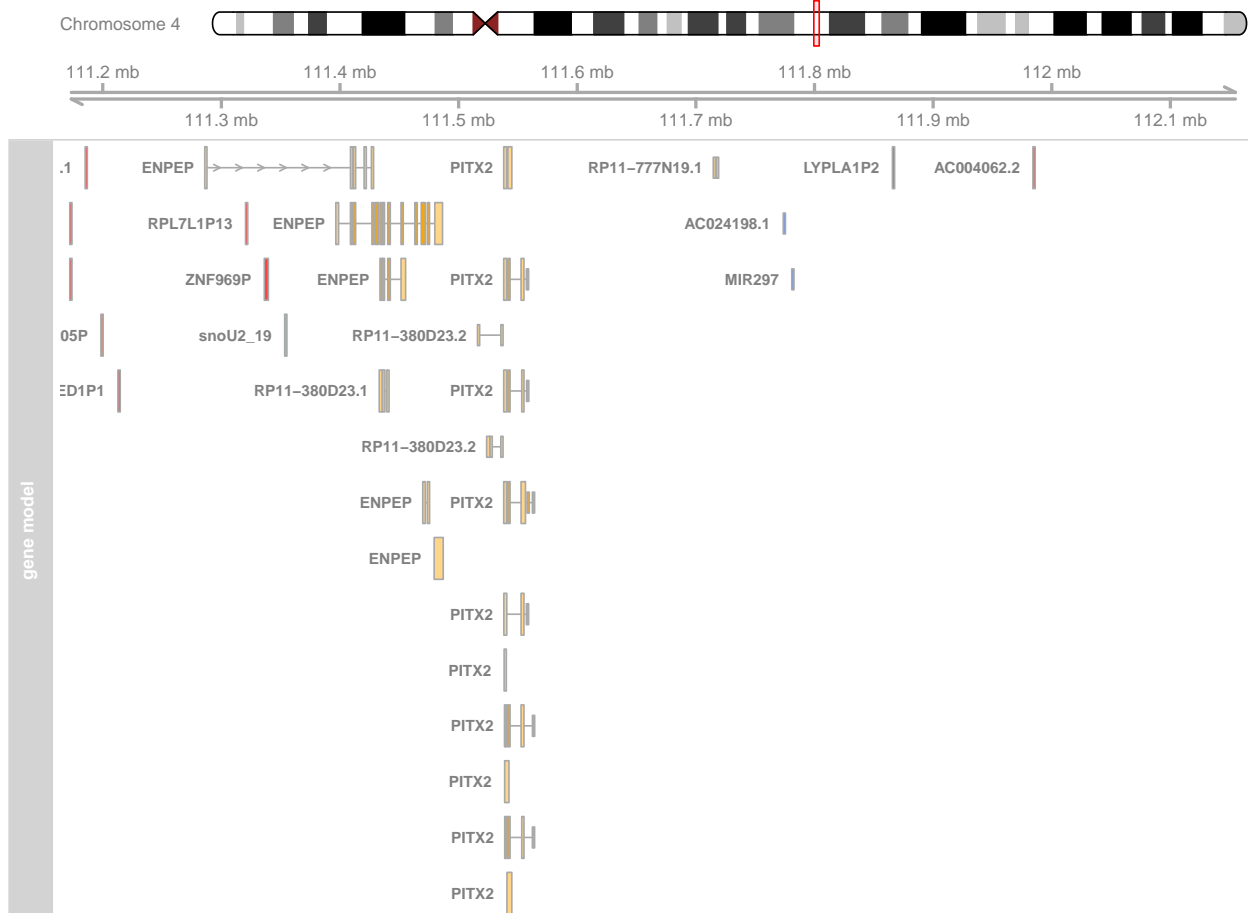
# get ideogram of the chromosome 4
itrack <- IdeogramTrack(genome = "hg19", chromosome = "4")

# get track for axis labeling
axtrack <- GenomeAxisTrack()

# get track with all genes with biomaRt annotation in the region
biomTrack <- BiomartGeneRegionTrack(genome = "hg19", name = "gene model",
                                     chromosome = 4,
                                     start = startRegion, end = endRegion,
                                     transcriptAnnotation = "symbol", frame=T)

# plot all tracks
plotTracks(list(itrack, axtrack, biomTrack),
            from = startRegion, to = endRegion)

```



8 Next week

- Primer on statistical modeling