A primer in Human Cardiovascular Genetics

dr. Sander W. van der Laan

Version 1.0.0 | last update: 2022-03-25

Contents

Chapter 1

About this primer

1.1 Introduction

Welcome to the *A primer in Human Cardiovascular Genetics* as part of the **Genetic Epidemiology** course. In the next few days we will use this GitBook to perform quality control (QC), executing a genome-wide association study (GWAS), annotating the GWAS results, and performing further downstream analyses. We will use data from the first release of the *Welcome Trust Case-Control Consortium (WTCCC)* and focus on coronary artery disease (CAD).

Unfortunately, during this course there is no time to perform imputation, but I will provide some pointers during the course as to how to do this with minimal coding/scripting experience. Likewise, this practical does not cover the aspects of meta-analyses of GWAS. But rest assured, I will add chapters on these subjects to a future version.

1.2 Background reading

Part of this is based on four great Nature Protocols from the Zondervan group at the Wellcome Center Human Genetics.

1. Zondervan KT *et al. Designing candidate gene and genome-wide case-control association studies.* Nat Protoc 2007.
2. Pettersson FH *et al. Marker selection for genetic case-control association studies.* Nat Protoc 2009.
3. Anderson CA *et al. Data QC in genetic case-control association studies.* Nat Protoc 2010.
4. Clarke GM *et al. Basic statistical analysis in genetic case-control studies.* Nat Protoc 2011.

An update on the community standards of QC for GWAS can be found here:

1. Laurie CC *et al. Quality control and quality assurance in genotypic data for genome-wide association studies.* Genet Epidemiol 2010.

With respect to imputation you should also get familiar with the following two works:

1. Marchini, J. and Howie, B. *Genotype imputation for genome-wide association studies.* Nat Rev Genet 2010
2. de Bakker PIW *et al. Practical aspects of imputation-driven meta-analysis of genome-wide association studies.* Hum Mol Genet 2008.
3. Winkler TW *et al. Quality control and conduct of genome-wide association meta-analyses.* Nat Protoc 2014.

1.3 Meet the Team

We work with a team of enthusiastic lecturers with experience in bioinformatics, GWAS, genetic analyses, Mendelian randomization, and epidemiology. This year the team consists of:

Sander W. van der Laan *Assistant professor*Course coordinators. w.vanderlaan-2@umcutrecht.nl | swvanderlaan

Charlotte Onland-Moret *Associate Professor*N.C.Onland@umcutrecht.nl | nconland

Jessica van Setten *Assistant professor*j.vansetten@umcutrecht.nl | j_vansetten

---

1.4 Final thoughts

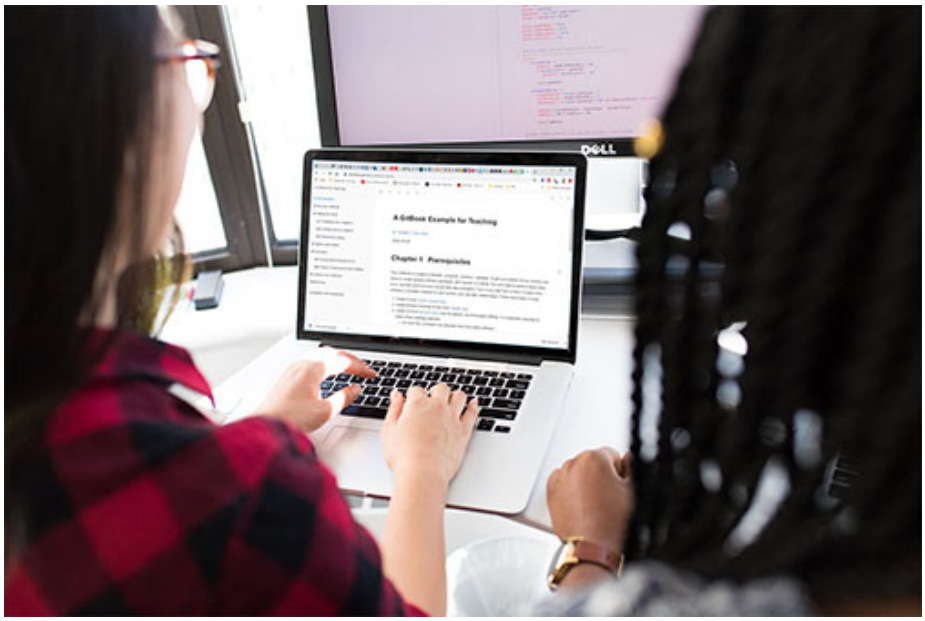
I can imagine this seems overwhelming, but trust me, you'll be okay. Just follow this practical, but also work on the questions asked during the lectures and in this practical. You'll learn by doing and at the end of the day, you can execute a GWAS independently.

Ready to start?

Your first point of action is to prepare your system for this course (Chapter 2).

Chapter 2

Prerequisites



2.1 Linux, macOS, and Windows

Most programs made to execute genetic epidemiology studies are developed for the Unix environment, for example Linux and macOS. So, they may not work as intended in a Windows environment. Windows does allow users to install a linux subsystem within Windows 10 and you can find the detail guide here.

However, I highly recommend to 1) either install a linux subsystem on your

Windows computer (for example a virtual machine with Ubuntu could work), or 2) switch to macOS in combination with homebrew. This will give you all the flexibility to use Unix-based programs for your genetic epidemiology work and at the same time you'll keep the advantage of a powerful computer with a user-friendly interface (either Windows or macOS).

> For this practical we use a Windows laptop with Ubuntu on a VirtualMachine. Therefore every command is intended for Linux/macOS, in other words Unix-systems.

2.2 Programs you need

You need few programs for this practical, or for your (future) genetic epidemiology work for that matter (**Table 1**).

Program	Link	Description
PLINK	https://www.cog-genomics.org/plink2/	PLINK is a free, open-source genetic analysis tool set, designed to perform a range of basic data parsing and quality control, as well as basic and large-scale analyses in a computationally efficient manner.
R	https://cran.r-project.org/	A program to perform statistical analysis and visualizations.
RStudio	https://www.rstudio.com	A user-friendly R-wrap-around for code editing, debugging, analyses, and visualization.
Homebrew	https://brew.sh	A great extension for Mac-users to install really useful programs that Apple didn't.

Table 1: Programs needed for genetic epidemiology.

All genetic analyses can be done in PLINK, even on your laptop, but with large datasets, for example UK Biobank size, it is better to switch to a high-performance computing cluster like we have available at the Utrecht Science Park.

Nowadays, a lot of people also use programs like SNPTEST, BOLT-LMM, GCTA, or regenie as alternatives to execute GWAS and downstream analyses, for example heritability estimation, Fst-calculation, and so on.

Mendelian randomization can be done either with the SMR or GSMR function from GCTA, or with R-packages, like `TwoSampleMR`.

2.3 The Terminal

For all the above programs, except RStudio, you will need the `Terminal`. This comes with every major operating system; on Windows it is called 'PowerShell', but let's not go there. And regardless, you will (have to start to) make your own scripts. The benefit of using scripts is that each step in your workflow is clearly stipulated and annotated, and it allows for greater reproducibility, easier troubleshooting, and scaling up to high-performance computer clusters.

Open the terminal, it should be on the left in the toolbar as a little black computer-monitor-like icon. Mac users can type `command + space` and type `terminal`, a terminal screen should open.

> From now on we will use little code blocks like the example to indicate a code you should type/copy-paste and hit enter. If a code is followed by a comment, it is indicated by a # - you don't need to copy-paste and execute this.

```
CODE BLOCK
```

```
CODE BLOCK # some comment here
```

2.3.1 Download the data

First, let's start by downloading the data you need for this course to your Desktop: LINK.

Alternatively, you could do this through this command. This will create a directory on your Desktop with the command `mkdir`. The `-v` flag indicates the program should be *verbose*, meaning it should tell you what it is doing.

```
mkdir -v ~/Desktop/practical/
```

```
wget "https://www.dropbox.com/sh/kumfwm7drt2flhp/AAB5n0OcUvJixI9pNiymx6-La?dl=0" -P ~/Desktop/pra
```

2.3.2 Navigating the Terminal

You can navigate around the computer through the terminal by typing `cd <path>`; `cd` stands for "change directory" and means "some_file_directory_you_want_to_go_to".

```
# For Linux/macOS Users
cd ~ # will bring you to your home directory
cd ../ # will bring you to the parent directory (up one level)
cd XXX # will bring you to the XXX directory
```

Let's navigate to the folder you just downloaded.

```
cd ~/Desktop/practical
```

Let's check out what is inside the directory, by listing (`ls`) its contents.

```
ls -lh
```

```
# For Linux/macOS Users
ls -l # shows files as list
ls -lh # shows files as list with human readable format
ls -lt # shows the files as list sorted by time edited
ls -lS # shows the files as list sorted by size
```

Adding the flags `-lh` will get you the contents of a directory in a list (`-l`) and make the size 'human-readable' (`-h`).

You can also count the number of files.

```
ls | wc -l
```

2.4 Installing some R packages

I tested this VirtualMachine and everything should be fine, except some libraries weren't there. We need to install them.

To be able to install certain `r`-packages, we need to install some Linux (Ubuntu) software. Type the following:

```
sudo apt-get install libcurl4 libcurl4-openssl-dev -y
```

```
sudo apt-get install libssl-dev
```

Now close the terminal window - really making sure that the terminal-program has quit.

Open a new terminal window and open `r` by simply typing `R` or `r`. You should install the following packages, and then you're good to go!

```
install.packages(c("httr", "usethis", "data.table", "devtools", "qqman", "CMplot", "til
devtools::install_github("kassambara/ggpubr")
```

You should load these packages too.

```
library("ggpubr")
library("httr")
library("usethis")
library("data.table")
library("devtools")
library("qqman")
library("CMplot")
library("tibble")
library("plotly")
library("dplyr")
```

All in all this may take some time, good moment to relax, review your notes, stretch your legs, or take a coffee.

Chapter 3

Basics of a Genome-Wide Association Study (GWAS)

Now that you understand a bit of the navigation in Unix-systems, we will continue with the practical. We will make use of a dummy dataset containing cases and controls. We will explain and execute the following steps:

- convert raw data to a more memory-efficient format
- apply extensive quality control
- perform association testing

3.1 Converting datasets

The format in which genotype data are returned to investigators varies among genome-wide SNP platforms and genotyping centers. Usually genotypes have been called by a genotyping center and returned in the standard `PED` and `MAP` file formats.

A `PED` file is a white space (space or tab)-delimited file in which each line represents one individual and the first six columns are mandatory and in the following order:

- 'Family ID',
- 'Individual ID',
- 'Paternal ID',
- 'Maternal ID',
- 'Sex (1=male, 2=female, 0=missing)', and
- 'Phenotype (1=unaffected, 2=affected, 0=missing)'.

The subsequent columns denote genotypes that can be any character (e.g., 1, 2, 3, 4 or A, C, G, T). Zero denotes a missing genotype. Each SNP must have two alleles (i.e., both alleles are either present or absent). The order of SNPs in the PED file is given in the MAP file, in which each line denotes a single marker and the four white-space–separated columns are chromosome (1–22, X, Y or 0 for unplaced), marker name (typically an rs number), genetic distance in Morgans (this can be fixed to 0) and base-pair position (bp units).

Let's start by using `PLINK` to converting the datasets to a lighter, binary form (a `BED`-file). BED files save data in a more memory- and time-efficient manner (binary files) to facilitate the analysis of large-scale data sets@purcell2007. PLINK creates a `.log` file (named `raw-GWA-data.log`) that details (among other information) the implemented commands, the number of cases and controls in the input files, any excluded data and the genotyping rate in the remaining data. This file is very useful for checking whether the software is successfully completing commands.

Make sure you are in the right directory. Do you remember how to get there?

```
cd ~/Desktop/practical
```

No worries for now: I've done this already for you!

```
plink --file rawdata/raw-GWA-data --make-bed --out rawdata/rawdata
```

3.2 Quality control

We are ready for some quality control and quality assurance, heavily inspired by Anderson *et al.*@anderson2010 and Laurie *et al.*@laurie2010. In general, we should check out a couple of things regarding the data quality on two levels:

1) samples
2) variants

So, we will investigate the following:

- Are the *sexes* based on genetic data matching the ones given by the phenotype file?
- Identify individuals that are outliers in terms of missing data (*call rate*) or heterozygosity rates. This could indicate a genotyping error or sample swap.
- Identify duplicated or related individuals.
- Identify individuals with divergent ancestry.
- What are the allele frequencies?

- What is the per-SNP call rate?
- In the case of a case-control study (which is the case here), we need to check differential missingness between cases and controls. By the way: you could extent this to for instance 'genotyping platform', or 'hospital of inclusion', if you think this might influence the genotyping experiment technically.

Right, on to step 1 of the QC in (Chapter @ref(gwas_basic_sample_qc)).

Chapter 4

Sample QC

Let's start with the per-sample quality control.

4.1 Sex

We need to identify of individuals with discordant sex information comparing phenotypic and genotypic data. Let's calculate the mean homozygosity rate across X-chromosome markers for each individual in the study.

```
plink --bfile rawdata/rawdata --check-sex --out rawdata/rawdata
```

This produces a file with the following columns:

- *FID* Family ID
- *IID* Within-family ID
- *PEDSEX* Sex code in input file
- *SNPSEX* Imputed sex code (1 = male, 2 = female, 0 = unknown)
- *STATUS* 'OK' if PEDSEX and SNPSEX match and are nonzero, 'PROBLEM' otherwise
- *F* Inbreeding coefficient, considering only X chromosome. Not present with 'y-only'.
- *YCOUNT* Number of nonmissing genotype calls on Y chromosome. Requires 'ycount'/'y-only'.

We need to get a list of individuals with discordant sex data.

```
cat rawdata/rawdata.sexcheck | awk '$5 =="STATUS" || $5 =="PROBLEM"'  > rawdata/rawdata.sexprobs.
```

Table 4.1: Sex issues

FID	IID	PEDSEX	SNPSEX	STATUS	F
772	772	2	0	PROBLEM	0.3084
853	853	2	0	PROBLEM	0.3666
1920	1920	2	0	PROBLEM	0.4066

Let's have a look at the results.

```
cat rawdata/rawdata.sexprobs.txt
```

```
sexissues <- data.table::fread(paste0(COURSE_loc,"/rawdata/rawdata.sexprobs.txt"))
knitr::kable(sexissues, caption = "Sex issues")
```

When the homozygosity rate (F) is more than 0.2, but less than 0.8, the genotype data are inconclusive regarding the sex of an individual and these are marked in column *SNPSEX* with a 0, and the column *STATUS* "PROBLEM".

Report the IDs of individuals with discordant sex information to those who conducted sex phenotyping. In situations in which discrepancy cannot be resolved, add the family ID (FID) and individual ID (IID) of the samples to a file named "fail-sexcheck-qc.txt" (one individual per line, tab delimited).

```
grep "PROBLEM" rawdata/rawdata.sexcheck | awk '{ print $1, $2}'  > rawdata/fail-sexche
```

4.2 Sample Callrates

Let's get an overview of the missing data per sample and per SNP.

```
plink --bfile rawdata/rawdata --missing --out rawdata/rawdata
```

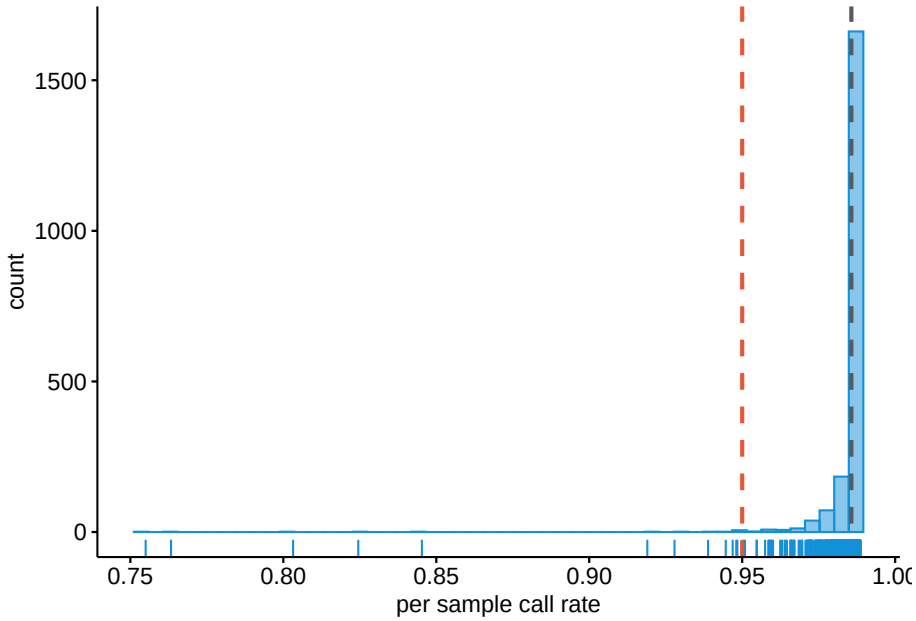
This produces two files, `rawdata/rawdata.imiss` and `rawdata/rawdata.lmiss`. In the .imiss file the *N_MISS* column denotes the number of missing SNPs, and the *F_MISS* column denotes the proportion of missing SNPs per individual.

```
raw_IMISS <- data.table::fread(paste0(COURSE_loc, "/rawdata/rawdata.imiss"))

raw_IMISS$callrate <- 1 - raw_IMISS$F_MISS

ggpubr::gghistogram(raw_IMISS, x = "callrate",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dash
```

```
                    rug = TRUE, bins = 50,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "per sample call rate") +
  geom_vline(xintercept = 0.95, linetype = "dashed",
                 color = "#E55738", size = 1)
```

```
## Warning: geom_vline(): Ignoring ‘mapping‘ because ‘xintercept‘ was provided.
```

```
## Warning: geom_vline(): Ignoring ‘data‘ because ‘xintercept‘ was provided.
```



The grey dashed line indicates the mean call rate, while the red dashed line indicates the threshold we had determined above.

4.3 Heterozygosity rate

To properly calculate heterozygosity rate and relatedness (identity-by-descent [IBD]) we need to do four things:

1) pre-clean the data to get a high-quality set,
2) of independent SNPs,
3) exclude long-range linkage disequilibrium (LD) blocks that bias with these calculations, and

4) exclude A/T and C/G SNPs as these may be ambivalent in interpretation when frequencies between cases and controls are close (MAF ± 0.45),
5) remove all non-autosomal SNPs.

We will use the following settings:

- remove A/T and C/G SNPs with the flag `--exclude rawdata/all.atcg.variants.txt`,
- call rate <1% with the flag `--geno 0.10`,
- Hardy-Weinberg Equilibrium (HWE) p-value > 1x10-3 with the flag `--hwe 1e-3`,
- and MAF>10% with the flag `--maf 0.10`,
- prune the data to only select independent SNPs (with low LD r^2) of one pair each with `r^2 = 0.2` with the flags `--indep-pairwise 100 10 0.2` and `--extract rawdata/raw-GWA-data.prune.in`,
- SNPs in long-range LD regions (for example: MHC chr 6 25.8-36Mb, chr 8 inversion 6-16Mb, chr17 40-45Mb, and a few more) with the flag `--exclude range rawdata/exclude_problematic_range.txt`,
- remove non-autosomal SNPs with the flag `--allow-no-sex --autosome`.

First, get a list of A/T and C/G SNPs.

```
cat rawdata/rawdata.bim | \
awk '($5 == "A" && $6 == "T") || ($5 == "T" && $6 == "A") || ($5 == "C" && $6 == "G")
> rawdata/all.atcg.variants.txt
```

Second, clean the data and get a list of independent SNPs.

```
plink --bfile rawdata/rawdata \
--allow-no-sex --autosome \
--maf 0.10 --geno 0.10 --hwe 1e-3 \
--indep-pairwise 100 10 0.2 \
--exclude range rawdata/exclude_problematic_range.txt \
--make-bed --out rawdata/rawdata.clean.temp
```

> Please note, we have create a dataset without taking into account LD structure. Thus the flag `--indep-pairwise 100 10 0.2` doesn't actually work. However, with real-data you can use it to prune out unwanted SNPs in high LD.

Third, exclude the pruned SNPs. Note, how we include a file to exclude high-LD for the purpose of the practical.

```
plink --bfile rawdata/rawdata.clean.temp \
--extract rawdata/raw-GWA-data.prune.in \
--make-bed --out rawdata/rawdata.clean.ultraclean.temp
```

Fourth, remove the A/T and C/G SNPs.

```
plink --bfile rawdata/rawdata.clean.ultraclean.temp \
--exclude rawdata/all.atcg.variants.txt \
--make-bed --out rawdata/rawdata.clean.ultraclean
```

> Please note, this dataset doesn't actually include this type of SNP, hence `rawdata/all.atcg.variants.txt` is empty! Again, you can use this command in real-data to exclude A/T and C/G SNPs.

Lastly, remove the temporary files.

```
rm -v rawdata/*.temp*
```

Finally, we can calculate the heterozygosity rate.

```
plink --bfile rawdata/rawdata.clean.ultraclean --het --out rawdata/rawdata.clean.ultraclean
```

This creates the file `rawdata/rawdata.clean.ultraclean.het`, in which the third column denotes the observed number of homozygous genotypes, O(Hom), and the fifth column denotes the number of nonmissing genotypes, N(NM), per individual. We can now calculate the observed heterozygosity rate per individual using the formula (N(NM) - O(Hom))/N(NM).

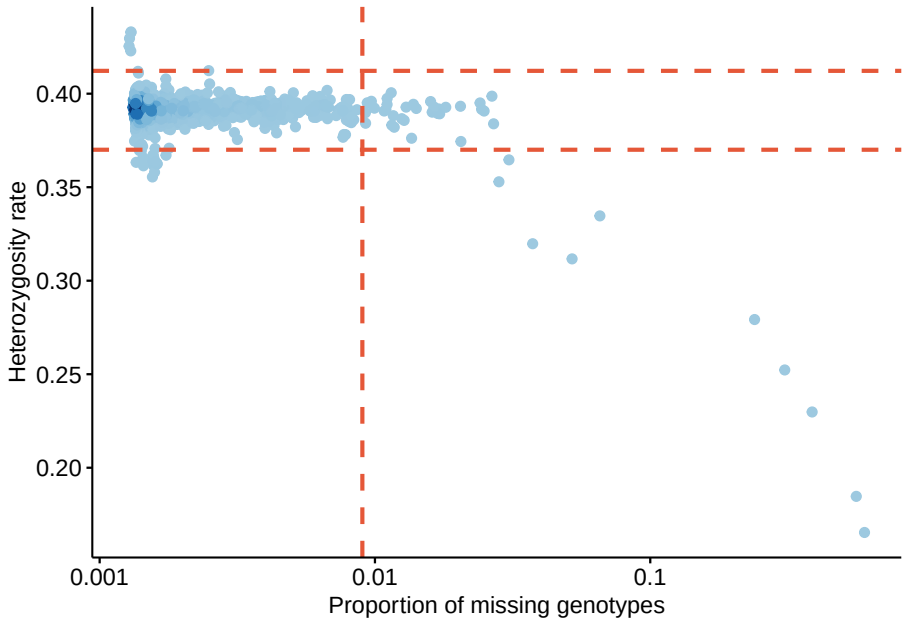
Often there is a correlation between heterozygosity rate and missing data. Thus, we should plot the observed heterozygosity rate per individual on the x-axis and the proportion of missing SNP, that is the 'SNP call rate', per individuals on the y-axis.

```
raw_HET <- data.table::fread(paste0(COURSE_loc, "/rawdata/rawdata.clean.ultraclean.het"))

raw_IMISS$logF_MISS = log10(raw_IMISS$F_MISS)
prop_miss = -1.522879

raw_HET$meanHet = (raw_HET$`N(NM)` - raw_HET$`O(HOM)`)/raw_HET$`N(NM)`
lower_meanHet = mean(raw_HET$meanHet) - (2*sd(raw_HET$meanHet))
upper_meanHet = mean(raw_HET$meanHet) + (2*sd(raw_HET$meanHet))

raw_IMISSHET = merge(raw_IMISS, raw_HET, by = "IID")
raw_IMISSHET$FID.y <- NULL
colnames(raw_IMISSHET)[colnames(raw_IMISSHET)=="FID.x"] <- "FID"

colors  <- densCols(raw_IMISSHET$logF_MISS, raw_IMISSHET$meanHet)
```

```
## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'
```

```
ggpubr::ggscatter(raw_IMISSHET, x = "logF_MISS", y = "meanHet",
                  color = colors,
                  xlab = "Proportion of missing genotypes", ylab = "Heterozygosity rate
  scale_x_continuous(labels=c("-3" = "0.001", "-2" = "0.01",
                              "-1" = "0.1", "0" = "1")) +
  geom_hline(yintercept = lower_meanHet, linetype = "dashed",
                color = "#E55738", size = 1) +
  geom_hline(yintercept = upper_meanHet, linetype = "dashed",
                color = "#E55738", size = 1) +
  geom_vline(xintercept = prop_miss, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning in if (color %in% names(data) & is.null(add.params$color))
## add.params$color <- color: the condition has length > 1 and only the first
## element will be used
```



Examine the plot to decide reasonable thresholds at which to exclude individuals based on elevated missing or extreme heterozygosity. We chose to exclude all individuals with a genotype failure rate >= 0.03 (vertical dashed line) and/or a heterozygosity rate ± 3 s.d. from the mean (horizontal dashed

lines). Add the FID and IID of the samples failing this QC to the file named `fail-imisshet-qc.txt`.

How would you create this file?

```
raw_IMISSHETsub = subset(raw_IMISSHET, logF_MISS > prop_miss | (meanHet < lower_meanHet | meanHet
                         select = c("FID", "IID"))
data.table::fwrite(raw_IMISSHETsub, paste0(COURSE_loc,"/rawdata/fail-raw_IMISSHETsub.txt"), sep =
```

4.4 Relatedness

We calculate Identity-by-Descent (IBS), to identify duplicated and related samples (**Table 2**). IBS is measured by calculating pi-hat, which is in essence the proportion of the DNA that a pair of samples share. To calculate this, we needed this ultraclean dataset, without low-quality SNPs and without high-LD regions. Now we are ready

Relation	**% DNA sharing**
Monozygotic twins	±100%
Parents/child	±50%
Sibling	±50%
Fraternal twins	±50%
Grandparent/grandchild	±25%
Aunt/Uncle/Niece/Nephew	±25%
Half-sibling	±25%
First-cousin	±12.5%
Half first-cousin	±6.25%
First-cousin once removed	±6.25%
Second-cousin	±3.13%
Second-cousin once removed	±1.56%

Table 2: Familial relations and % DNA shared.

```
plink --bfile rawdata/rawdata.clean.ultraclean --genome --out rawdata/rawdata.clean.ultraclean
```

We can now identify all pairs of individuals with an IBD > 0.185. The code looks at the individual call rates stored in rawdata.imiss and outputs the IDs of the individual with the lowest call rate to 'fail-IBD-QC.txt' for subsequent removal.

Table 4.3: Failed IBD and callrate

V1	V2
1952	1952
1953	1953
1954	1954
1955	1955
1957	1957
1959	1959
1961	1961
1963	1963
1965	1965
1967	1967
1969	1969
1971	1971
1973	1973
1975	1975

```
cd rawdata

perl ../scripts/run-IBD-QC.pl rawdata rawdata.clean.ultraclean

cd ..
```

```
ibdcallissues <- data.table::fread(paste0(COURSE_loc,"/rawdata/fail-IBD-QC.txt"))
knitr::kable(ibdcallissues, caption = "Failed IBD and callrate")
```

4.5 Ancestral background

4.5.1 HapMap 3

We will project our data to a reference, in this example HapMap Phase II (HapMap3), which includes individuals from four distinct global populations, but it could also be 1000G phase 1. Or any other reference depending on the dataset.

To this end we will merge our data with HapMap3. The alleles at each marker must be aligned to the same DNA strand to allow our data to merge correctly. Because not all SNPs are required for this analysis, A->T and C->G SNPs, which are more difficult to align, can be omitted.

Let's start by creating a new BED file, excluding from the GWA data those

SNPs that do not feature in the genotype data of the four original HapMap3 populations.

```
plink --bfile rawdata/rawdata --extract reference/hapmap3r2_CEU.CHB.JPT.YRI.no-at-cg-snps.txt --m
```

Now, let's try to merge `rawdata/rawdata.hm3` with the HapMap data and extract the pruned SNP set from above.

```
plink --bfile rawdata/rawdata.hm3 --bmerge reference/hapmap3r2_CEU.CHB.JPT.YRI.founders.no-at-cg-
```

You probably get an error like below:

```
Error: 59 variants with 3+ alleles present.
* If you believe this is due to strand inconsistency, try --flip with
  rawdata/rawdata.hapmap3r2.pruned-merge.missnp.
  (Warning: if this seems to work, strand errors involving SNPs with A/T or C/G
  alleles probably remain in your data.  If LD between nearby SNPs is high,
  --flip-scan should detect them.)
* If you are dealing with genuine multiallelic variants, we recommend exporting
  that subset of the data to VCF (via e.g. '--recode vcf'), merging with
  another tool/script, and then importing the result; PLINK is not yet suited
  to handling them.
```

Because all A->T and C->G SNPs have been removed before undertaking this analysis, all other SNPs that are discordant for DNA strands between the two data sets are listed in the `rawdata.hapmap3r2.pruned-merge.missnp` file. To align the strands across the data sets and successfully complete the merge, we can do the following:

```
plink --bfile rawdata/rawdata --extract reference/hapmap3r2_CEU.CHB.JPT.YRI.no-at-cg-snps.txt --f
```

And repeat this:

```
plink --bfile rawdata/rawdata.hm3 --bmerge reference/hapmap3r2_CEU.CHB.JPT.YRI.founders.no-at-cg-
```

Let's not be lazy and clean this dataset too.

```
plink --bfile rawdata/rawdata.hapmap3r2.pruned \
--allow-no-sex --autosome \
--maf 0.10 --geno 0.10 --hwe 1e-3 \
--indep-pairwise 100 10 0.2 \
--exclude range rawdata/exclude_problematic_range.txt \
--make-bed --out rawdata/rawdata.hapmap3r2.pruned.clean
```

4.5.2 Principal Component Analysis

Using a Principal Component Analysis (PCA) we can reduce the dimensions of the data, and project the "ancestral distances". In other words, the principal component 1 (the first dimension) and principal component 2 (the second dimension) which will capture most of the variation in the data and represent how much each sample is alike the next.

First, we make a copy of the BIM and FAM-files.

```
cp -v rawdata/rawdata.hapmap3r2.pruned.bim rawdata/rawdata.hapmap3r2.pruned.pedsnp
cp -v rawdata/rawdata.hapmap3r2.pruned.fam rawdata/rawdata.hapmap3r2.pruned.pedind
```

4.5.2.1 Installing EIGENSOFT

Now, we are ready to perform the PCA using `smartPCA`. For this `EIGENSOFT` needs to be installed. Unfortunately, this doesn't work on this VirtualMachine you are working on - you need `gsl`, `openblas` and `llvm` to make it work.

Installing EIGENSOFT

I am still sharing the code you'll need - you could try this on your personal MacBook for instance.

```
mkdir -v $HOME/git
cd $HOME/git
git clone https://github.com/DReichLab/EIG.git
cd EIG/src
make
make install
```

Executing smartPCA

Should you run this on your personal laptop, be aware it will take a few minutes to do so - perfect moment for a cup of coffee or to stretch your legs.

```
perl ~/git/EIG/bin/smartpca.perl \
-i rawdata/rawdata.hapmap3r2.pruned.bed \
-a rawdata/rawdata.hapmap3r2.pruned.pedsnp \
-b rawdata/rawdata.hapmap3r2.pruned.pedind \
-k 10 \
-o rawdata/rawdata.hapmap3r2.pruned.pca \
-p rawdata/rawdata.hapmap3r2.pruned.plot \
-e rawdata/rawdata.hapmap3r2.pruned.eval \
-l rawdata/rawdata.hapmap3r2.pruned.log \
-m 5 \
```

```
-t 10 \
-s 6.0 \
-w reference/hapmap3r2_CEU.CHB.JPT.YRI-pca-populations.txt
```

See below an explanation of the above commands:

`../bin/smartpca.perl`

-i example.geno : genotype file in any format (see ../CONVERTF/README) -a example.snp : snp file in any format (see ../CONVERTF/README) -b example.ind : indiv file in any format (see ../CONVERTF/README) -k k : (Default is 10) number of principal components to output -o example.pca : output file of principal components. Individuals removed as outliers will have all values set to 0.0 in this file. -p example.plot : prefix of output plot files of top 2 principal components. (labeling individuals according to labels in indiv file) -e example.eval : output file of all eigenvalues -l example.log : output logfile -m maxiter : (Default is 5) maximum number of outlier removal iterations. To turn off outlier removal, set -m 0. -t topk : (Default is 10) number of principal components along which to remove outliers during each outlier removal iteration. -s sigma : (Default is 6.0) number of standard deviations which an individual must exceed, along one of topk top principal components, in order to be removed as an outlier.

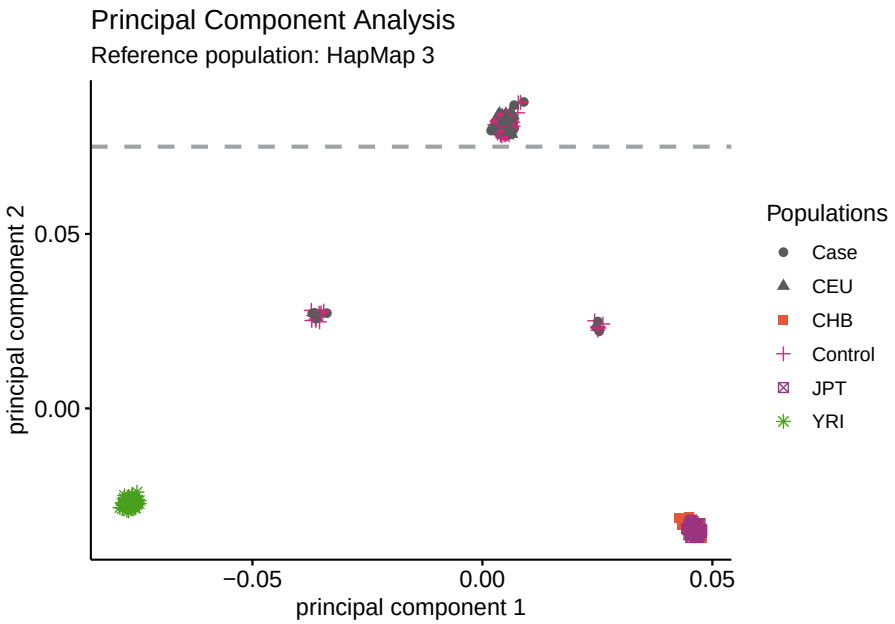
OPTIONAL FLAGS: -w poplist : compute eigenvectors using populations in poplist only, where poplist is an ASCII file with one population per line -y plotlist : output plot will include populations in plotlist only, where plotlist is an ASCII file with one population per line -z badsnpname : list of SNPs which should be excluded from the analysis -q YES/NO : If set to YES, assume that there is a single population and the population field contains real-valued phenotypes. (Corresponds to qtmode parameter in smartpca program.) The default value for this parameter is NO.

> NOTE: I made sure that in your download the results from this analysis are available for usage. That is:
>
> - rawdata/rawdata.hapmap3r2.pruned.evec
> - rawdata/rawdata.hapmap3r2.pruned.par

4.5.2.2 PCA plotting

Now that we have calculated PCs, we can start plotting them. Let's create a scatter diagram of the first two principal components, including all individuals in the file `rawdata.hapmap3r2.pruned.pca.evec` (the first and second principal components are columns 2 and 3, respectively). Use the data in column 4 to color the points according to sample origin. An R script for creating this plot (`scripts/plot-pca-results.Rscript`) is provided (although any standard graphing software can be used).

```
PCA <- data.table::fread(paste0(COURSE_loc,"/rawdata/rawdata.hapmap3r2.pruned.pca.evec'

# Case/Control -> black, pch = "+"
# CEU = 3 -> red, pch = 20
# CHB = 4 -> pink, pch = 20
# JPT = 5 -> purple, pch = 20
# YRI = 6 -> green, pch = 20
PCA$V12[PCA$V12 == "Case"] <- "Case" #595A5C
PCA$V12[PCA$V12 == "Control"] <- "Control" #595A5C
PCA$V12[PCA$V12 == "3"] <- "CEU" #E55738
PCA$V12[PCA$V12 == "4"] <- "CHB" #D5267B
PCA$V12[PCA$V12 == "5"] <- "JPT" #9A3480
PCA$V12[PCA$V12 == "6"] <- "YRI" #49A01D



PCAplot <- ggpubr::ggscatter(PCA, x = "V2", y = "V3",
                             color = "V12",
                             palette = c("#595A5C", "#595A5C", "#E55738", "#D5267B", "#
                             shape = "V12",
                             xlab = "principal component 1", ylab = "principal componer
  geom_hline(yintercept = 0.075, linetype = "dashed",
               color = "#A2A3A4", size = 1)

  ggpubr::ggpar(PCAplot,
               title = "Principal Component Analysis",
               subtitle = "Reference population: HapMap 3",
               legend.title = "Populations", legend = "right")
```



Derive PC1 and PC2 thresholds so that only individuals who match the given ancestral population are included. For populations of European descent, this will be either the CEU or TSI HapMap3 individuals. Here, we chose to exclude all individuals with a second principal component score less than 0.072.

Write the FID and IID of these individuals to a file called `fail-ancestry-QC.txt`.

```
cat rawdata/rawdata.hapmap3r2.pruned.pca.evec | tail -n +2 | \
awk '$3 < 0.075' | awk '{ print $1 }' | awk -F":" '{ print $1, $2 }' > rawdata/fail-ancestry-QC.t
```

Choosing which thresholds to apply (and thus which individuals to remove) is not a straightforward process. The key is to remove those individuals with greatly divergent ancestry, as these samples introduce the most bias to the study. Identification of more fine-scale ancestry can be conducted by using less divergent reference samples (*e.g.*, within Europe, stratification could be identified using the CEU, TSI (Italian), GBR (British), FIN (Finnish) and IBS (Iberian) samples from the 1,000 Genomes Project (http://www.1000genomes.org/)). Robust identification of fine-scale population structure often requires the construction of many (2–10) principal components.

4.5.3 1000G phase 1

You could do the above again but now with projecting the 1000G phase 1 populations. The all the 1000G phase 1 data is provided as tutorial data (), as

well as a subset including *only* the variants in our `rawdata`. You can try and run the PCA with 1000G and project the results – let's do that in our spare time and continue for now with the QC based on HM3. But please, do show us the results tomorrow ... :-) For your convenience there are some codes below which you may need.

Get a list of relevant variants.

```
cat rawdata/rawdata.bim | grep "rs" > rawdata/all.variants.txt
```

Extract those from the 1000G phase 1 data.

```
plink --bfile reference/1kg_phase1_all/1kg_phase1_all --extract rawdata/all.variants.t
```

Get a list of A/T and C/G variants from 1000G to exclude.

```
cat reference/1kg_phase1_all/1kg_phase1_raw.bim | \
awk '($5 == "A" && $6 == "T") || ($5 == "T" && $6 == "A") || ($5 == "C" && $6 == "G")
> reference/1kg_phase1_all/all.1kg.atcg.variants.txt
```

Exclude those A/T and C/G variants in both datasets.

```
plink --bfile reference/1kg_phase1_all/1kg_phase1_raw --exclude reference/1kg_phase1_a
```

```
plink --bfile rawdata/rawdata --exclude reference/1kg_phase1_all/all.1kg.atcg.variants
```

Try and merge the data while extracting the pruned SNP-set.

```
plink --bfile rawdata/rawdata_1kg_phase1_raw_no_atcg --bmerge reference/1kg_phase1_all/
```

There probably is an error ...

```
Error: 72 variants with 3+ alleles present.
* If you believe this is due to strand inconsistency, try --flip with
  rawdata/rawdata.1kg_phase1.pruned-merge.missnp.
  (Warning: if this seems to work, strand errors involving SNPs with A/T or C/G
  alleles probably remain in your data.  If LD between nearby SNPs is high,
  --flip-scan should detect them.)
* If you are dealing with genuine multiallelic variants, we recommend exporting
  that subset of the data to VCF (via e.g. '--recode vcf'), merging with
  another tool/script, and then importing the result; PLINK is not yet suited
  to handling them.
See https://www.cog-genomics.org/plink/1.9/data#merge3 for more discussion.
```

So let's flip some variants.

```
plink --bfile rawdata/rawdata --exclude reference/1kg_phase1_all/all.1kg.atcg.variants.txt --flip
```

Let's try and merge the data while extracting the pruned SNP-set.

```
plink --bfile rawdata/rawdata_1kg_phase1_raw_no_atcg --bmerge reference/1kg_phase1_all/1kg_phase1
```

There still is an error – there are multi-allelic variants present which PLINK can't handle.

```
Error: 14 variants with 3+ alleles present.
* If you believe this is due to strand inconsistency, try --flip with
  rawdata/rawdata.1kg_phase1.pruned-merge.missnp.
  (Warning: if this seems to work, strand errors involving SNPs with A/T or C/G
  alleles probably remain in your data.  If LD between nearby SNPs is high,
  --flip-scan should detect them.)
* If you are dealing with genuine multiallelic variants, we recommend exporting
  that subset of the data to VCF (via e.g. '--recode vcf'), merging with
  another tool/script, and then importing the result; PLINK is not yet suited
  to handling them.
See https://www.cog-genomics.org/plink/1.9/data#merge3 for more discussion.
```

Let's just remove these multi-allelic variants.

```
plink --bfile rawdata/rawdata_1kg_phase1_raw_no_atcg --exclude rawdata/rawdata.1kg_phase1.pruned-
```

Now we should be able to merge the data...

```
plink --bfile rawdata/rawdata_1kg_phase1_raw_no_atcg_bi --bmerge reference/1kg_phase1_all/1kg_pha
```

That worked! Let's run a PCA.

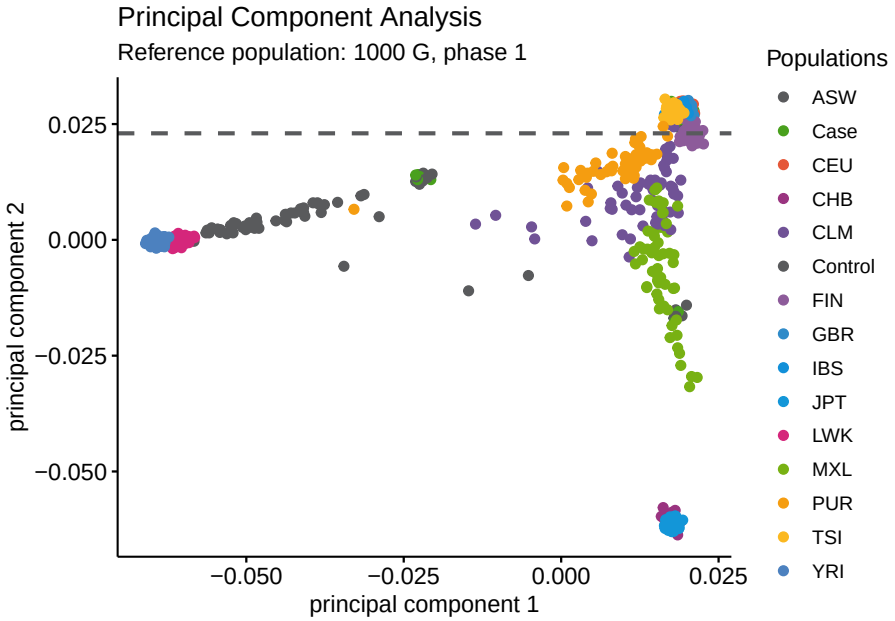
```
cp -v rawdata/rawdata.1kg_phase1.pruned.bim rawdata/rawdata.1kg_phase1.pruned.pedsnp
cp -v rawdata/rawdata.1kg_phase1.pruned.fam rawdata/rawdata.1kg_phase1.pruned.pedind

perl ~/git/EIG/bin/smartpca.perl \
-i rawdata/rawdata.1kg_phase1.pruned.bed \
-a rawdata/rawdata.1kg_phase1.pruned.pedsnp \
-b rawdata/rawdata.1kg_phase1.pruned.pedind \
-k 10 \
-o rawdata/rawdata.1kg_phase1.pruned.pca \
-p rawdata/rawdata.1kg_phase1.pruned.plot \
```

```
-e rawdata/rawdata.1kg_phase1.pruned.eval \
-l rawdata/rawdata.1kg_phase1.pruned.log \
-m 5 \
-t 10 \
-s 6.0 \
-w reference/1kg_phase1_all/1kg-pca-populations.txt
```

And we can try to plot this result as well.

```
PCA_1kG <- data.table::fread(paste0(COURSE_loc,"/rawdata/rawdata.1kg_phase1.pruned.pca

# Population    Description Super population    Code    Counts
# ASW   African Ancestry in Southwest US                                    AFR   4     #49A
# CEU   Utah residents with Northern and Western European ancestry  EUR 7     #E55738
# CHB   Han Chinese in Bejing, China                                    EAS   8     #9A3
# CHS   Southern Han Chinese, China                                 EAS 9     #705296
# CLM   Colombian in Medellin, Colombia                             MR  10  #8D5B9A
# FIN   Finnish in Finland                                          EUR 12  #2F8BC9
# GBR   British in England and Scotland                             EUR 13  #1290D9
# IBS   Iberian populations in Spain                                    EUR   16  #1396D
# JPT   Japanese in Tokyo, Japan                                    EAS   18  #D5267
# LWK   Luhya in Webuye, Kenya                                      AFR 20  #78B113
# MXL   Mexican Ancestry in Los Angeles, California                 AMR 22  #F59D10
# PUR   Puerto Rican in Puerto Rico                                 AMR 25  #FBB820
# TSI   Toscani in Italy                                                EUR   27  #4C81B
# YRI   Yoruba in Ibadan, Nigeria                                       AFR   28  #C5D22

PCA_1kG$V12[PCA_1kG$V12 == "Case"] <- "Case"
PCA_1kG$V12[PCA_1kG$V12 == "Control"] <- "Control"
PCA_1kG$V12[PCA_1kG$V12 == "4"] <- "ASW"
PCA_1kG$V12[PCA_1kG$V12 == "7"] <- "CEU"
PCA_1kG$V12[PCA_1kG$V12 == "8"] <- "CHB"
PCA_1kG$V12[PCA_1kG$V12 == "9"] <- "CHS"
PCA_1kG$V12[PCA_1kG$V12 == "10"] <- "CLM"
PCA_1kG$V12[PCA_1kG$V12 == "12"] <- "FIN"
PCA_1kG$V12[PCA_1kG$V12 == "13"] <- "GBR"
PCA_1kG$V12[PCA_1kG$V12 == "16"] <- "IBS"
PCA_1kG$V12[PCA_1kG$V12 == "18"] <- "JPT"
PCA_1kG$V12[PCA_1kG$V12 == "20"] <- "LWK"
PCA_1kG$V12[PCA_1kG$V12 == "22"] <- "MXL"
PCA_1kG$V12[PCA_1kG$V12 == "25"] <- "PUR"
PCA_1kG$V12[PCA_1kG$V12 == "27"] <- "TSI"
PCA_1kG$V12[PCA_1kG$V12 == "28"] <- "YRI"

PCA_1kGplot <- ggpubr::ggscatter(PCA_1kG, x = "V2", y = "V3",
```

```
                                color = "V12",
                                palette = c("#595A5C", "#49A01D", "#E55738", "#9A3480", "#705296
                                xlab = "principal component 1", ylab = "principal component 2")
  geom_hline(yintercept = 0.023, linetype = "dashed",
             color = "#595A5C", size = 1)

  ggpubr::ggpar(PCA_1kGplot,
             title = "Principal Component Analysis",
             subtitle = "Reference population: 1000 G, phase 1",
             legend.title = "Populations", legend = "right")
```



In a similar fashion as in the above with the HapMap3 reference, you could remove the samples below the threshold.

4.6 Removing samples

Finally! We have a list of samples of poor quality or divergent ancestry, and duplicated or related samples. We should remove these. Let's collect all IDs from our `fail-*`-files into a single file.

```
cat rawdata/fail-* | sort -k1 | uniq > rawdata/fail-qc-inds.txt
```

This new file should now contain a list of unique individuals failing the previous QC steps which we want to remove.

```
plink --bfile rawdata/rawdata --remove rawdata/fail-qc-inds.txt --make-bed --out rawdat
```

Chapter 5

Per-SNP QC

Now that we removed samples, we can focus on low-quality variants.

5.1 SNP call rates

We start by calculating the missing genotype rate for each SNP, in other words the per-SNP call rate.

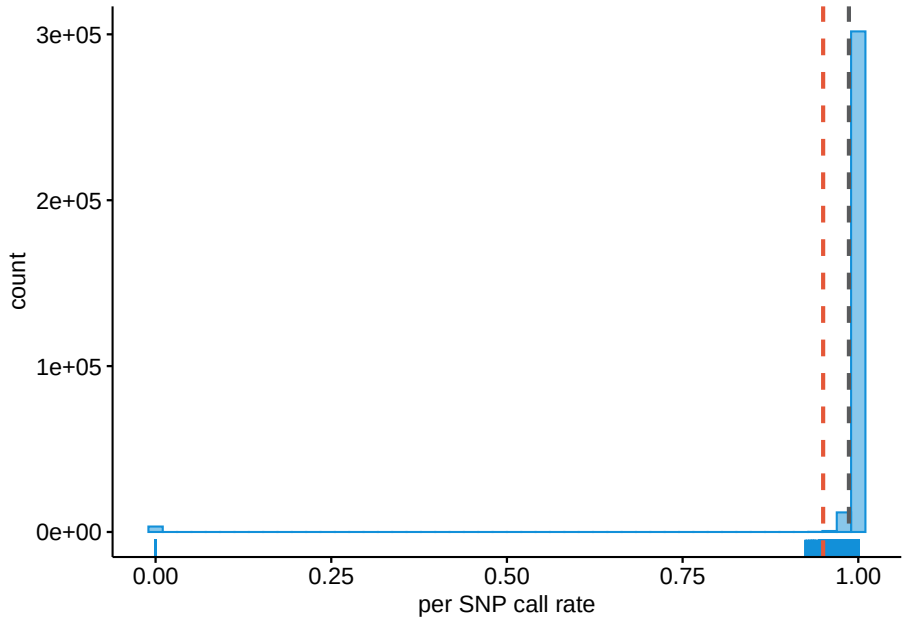
```
plink --bfile rawdata/clean_inds_data --missing --out rawdata/clean_inds_data
```

Let's visualize the results to identify a threshold for extreme genotype failure rate. We chose a callrate threshold of 3%, but it's arbitrary and depending on the dataset and the number of samples.

```
clean_LMISS <- data.table::fread(paste0(COURSE_loc, "/rawdata/clean_inds_data.lmiss"))

clean_LMISS$callrate <- 1 - clean_LMISS$F_MISS

ggpubr::gghistogram(clean_LMISS, x = "callrate",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dashed", size
                    rug = TRUE, bins = 50,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "per SNP call rate") +
  geom_vline(xintercept = 0.95, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```

```
## Warning: geom_vline(): Ignoring ‘data‘ because ‘xintercept‘ was provided.
```



5.2 Differential SNP call rates

There could also be differences in genotype call rates between cases and controls. It is very important to check for this because these differences could lead to spurious associations. We can test all markers for differences in call rate between cases and controls, or based on

```
plink --bfile rawdata/clean_inds_data --test-missing --out rawdata/clean_inds_data
```

Let's collect all the SNPs ith a significantly different ($P < 0.00001$) missing data rate between cases and controls.

```
cat rawdata/clean_inds_data.missing | awk '$5 < 0.00001' | awk '{ print $2 }' > rawdata
```

5.3 Allele frequencies

We should also get an idea on what the allele frequencies are in our dataset. Low frequent SNPs should probably be excluded, as these are uninformative when monomorphic (allele frequency = 0), or they may lead to spurious associations.

```
plink --bfile rawdata/clean_inds_data --freq --out rawdata/clean_inds_data
```

Let's also plot these data. You can view the result below, and type over the code to do it yourself.

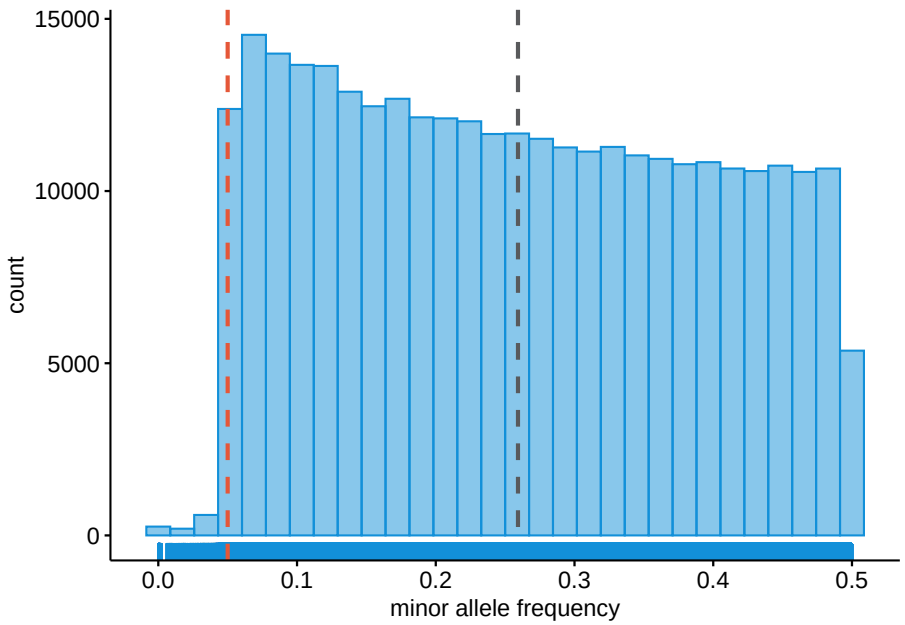
```
clean_FREQ <- data.table::fread(paste0(COURSE_loc, "/rawdata/clean_inds_data.frq"))
ggpubr::gghistogram(clean_FREQ, x = "MAF",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dashed", size
                    rug = TRUE,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "minor allele frequency") +
  geom_vline(xintercept = 0.05, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```

```
## Warning: geom_vline(): Ignoring `data` because `xintercept` was provided.
```

```
## Warning: Removed 3286 rows containing non-finite values (stat_bin).
```



A note on allele coding

Oh, one more thing about alleles.

`PLINK` codes alleles as follows:

A1 = minor allele, the least frequent allele A2 = major allele, the most frequent allele

And when you use `PLINK` the flag `--freq` or `--maf` is always relative to the A1-allele, as is the odds ratio (OR) or effect size (beta).

However, `SNPTEST` makes use of the so-called OXFORD-format, this codes alleles as follows:

A = the 'other' allele B = the 'coded' allele

When you use `SNPTEST` it will report the allele frequency as `CAF`, in other words the *coded allele frequency*, and the effect size (beta) is always relative to the B-allele. This means, `CAF` *could* be the `MAF`, or *minor allele frequency*, but this is **not** a given.

In other words, always make sure what the allele-coding of a given program, be it `PLINK`, `SNPTEST`, `GCTA`, et cetera, is! I cannot stress this enough. Ask yourself: 'what is the allele frequency refering to?', 'the effect size is relative to...?'.

Right, let's continue.

5.4 Hardy-Weinberg Equilibrium

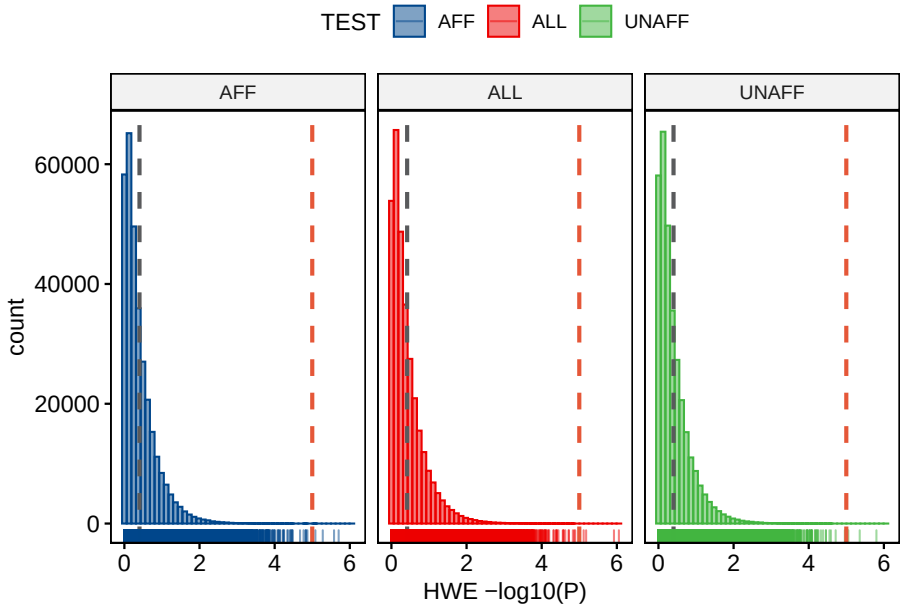
Because we are performing a case-control genome-wide association study, we probably expect some differences in Hardy-Weinberg Equilibrium (HWE), but extreme deviations are probably indicative of genotyping errors.

```
plink --bfile rawdata/clean_inds_data --hardy --out rawdata/clean_inds_data
```

Let's also plot these data. You can view the result below, and type over the code to do it yourself.

```
clean_HWE <- data.table::fread(paste0(COURSE_loc, "/rawdata/clean_inds_data.hwe"))
clean_HWE$logP <- -log10(clean_HWE$P)

ggpubr::gghistogram(clean_HWE, x = "logP",
                    add = "mean",
                    add.params = list(color = "#595A5C", linetype = "dashed", size = 1)
                    rug = TRUE,
                    # color = "#1290D9", fill = "#1290D9",
                    color = "TEST", fill = "TEST",
                    palette = "lancet",
```

```
                    facet.by = "TEST",
                    bins = 50,
                    xlab = "HWE -log10(P)") +
  geom_vline(xintercept = 5, linetype = "dashed",
                 color = "#E55738", size = 1)
```



5.5 Final SNP QC

We are ready to perform the final QC. After inspectig the graphs we will filter on a MAF < 0.01, call rate < 0.05, and HWE < 0.00001, in addition those SNPs that failed the differential call rate test will be removed.

```
plink --bfile rawdata/clean_inds_data --exclude rawdata/fail-diffmiss-qc.txt --maf 0.01 --geno 0.
```

Chapter 6

Genome-wide association study

Now that you have learned how to perform QC, you can easily run a GWAS and execute some downstream visualisation and analyses. Let's do this with a dummy dataset.

6.1 Exploring the data

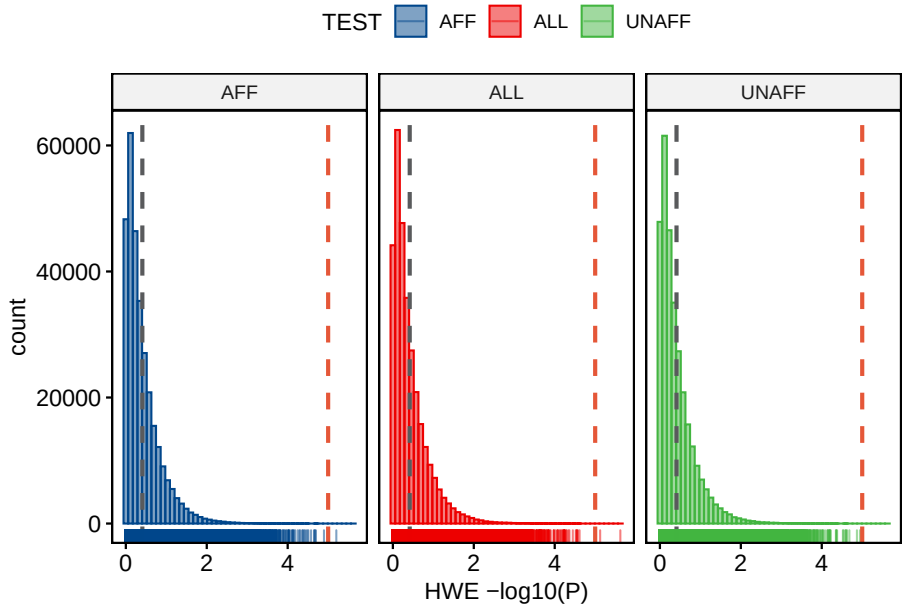
Even though someone says that the QC was done, it is still wise and good practice to run some of the commands above to get a 'feeling' about the data. So let's do this.

```
plink --bfile gwas/gwa --freq --out gwas/gwa

plink --bfile gwas/gwa --missing --out gwas/gwa

plink --bfile gwas/gwa --hardy --out gwas/gwa
```

Let's visualise the results.

```
gwas_HWE <- data.table::fread(paste0(COURSE_loc, "/gwas/gwa.hwe"))
gwas_FRQ <- data.table::fread(paste0(COURSE_loc, "/gwas/gwa.frq"))
gwas_IMISS <- data.table::fread(paste0(COURSE_loc, "/gwas/gwa.imiss"))
gwas_LMISS <- data.table::fread(paste0(COURSE_loc, "/gwas/gwa.lmiss"))

gwas_HWE$logP <- -log10(gwas_HWE$P)
```

```
ggpubr::gghistogram(gwas_HWE, x = "logP",
                    add = "mean",
                    add.params = list(color = "#595A5C", linetype = "dashed", size = 1)
                    rug = TRUE,
                    # color = "#1290D9", fill = "#1290D9",
                    color = "TEST", fill = "TEST",
                    palette = "lancet",
                    facet.by = "TEST",
                    bins = 50,
                    xlab = "HWE -log10(P)") +
  geom_vline(xintercept = 5, linetype = "dashed",
                color = "#E55738", size = 1)
```

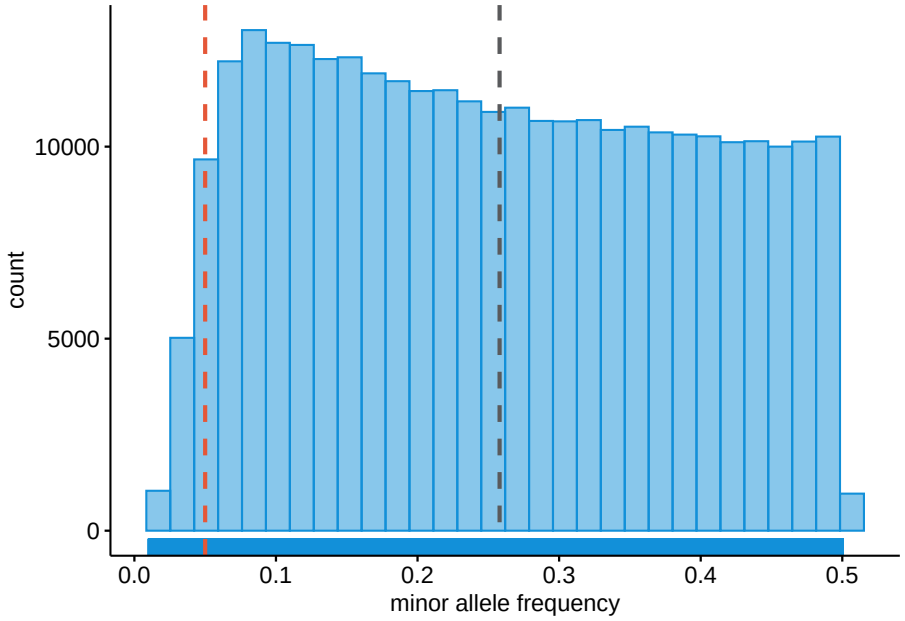



```
ggpubr::gghistogram(gwas_FRQ, x = "MAF",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dash
                    rug = TRUE,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "minor allele frequency") +
  geom_vline(xintercept = 0.05, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning: Using ‘bins = 30‘ by default. Pick better value with the argument
## ‘bins‘.
```

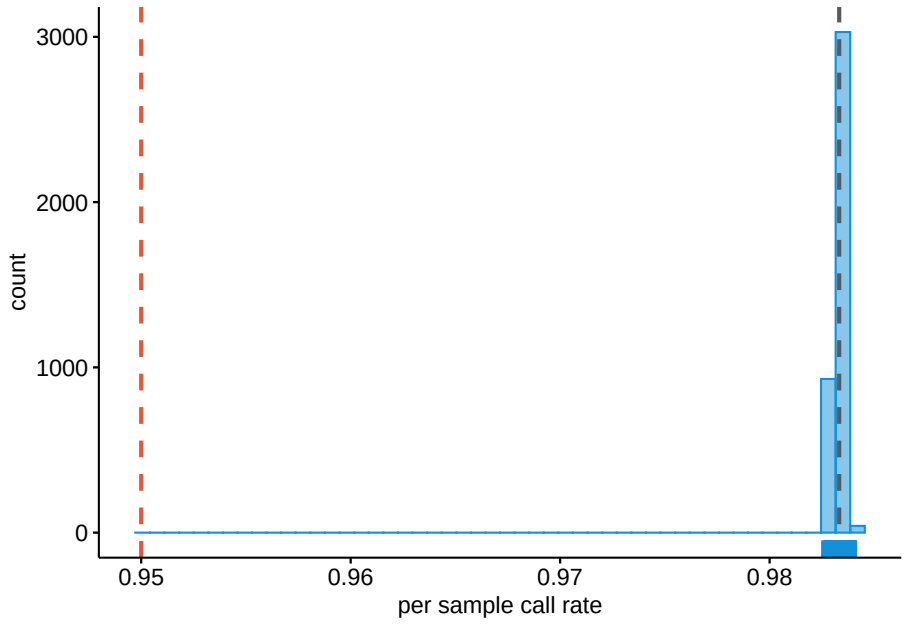
```
## Warning: geom_vline(): Ignoring ‘mapping‘ because ‘xintercept‘ was provided.
```

```
## Warning: geom_vline(): Ignoring ‘data‘ because ‘xintercept‘ was provided.
```



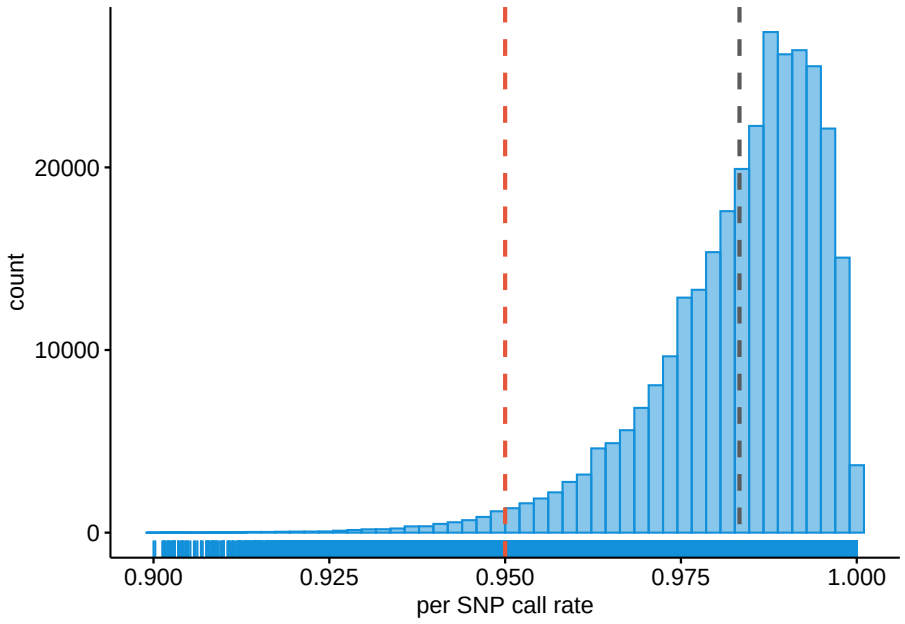

```
gwas_IMISS$callrate <- 1 - gwas_IMISS$F_MISS

ggpubr::gghistogram(gwas_IMISS, x = "callrate",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dashed", size
                    rug = TRUE, bins = 50,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "per sample call rate") +
  geom_vline(xintercept = 0.95, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning: geom_vline(): Ignoring ‘mapping‘ because ‘xintercept‘ was provided.
## geom_vline(): Ignoring ‘data‘ because ‘xintercept‘ was provided.
```




```
gwas_LMISS$callrate <- 1 - gwas_LMISS$F_MISS

ggpubr::gghistogram(gwas_LMISS, x = "callrate",
                    add = "mean", add.params = list(color = "#595A5C", linetype = "dash
                    rug = TRUE, bins = 50,
                    color = "#1290D9", fill = "#1290D9",
                    xlab = "per SNP call rate") +
  geom_vline(xintercept = 0.95, linetype = "dashed",
                color = "#E55738", size = 1)
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
## geom_vline(): Ignoring `data` because `xintercept` was provided.
```



6.2 Genetic models

A simple chi-square test of association can be done.

```
plink --bfile gwas/gwa --model --out gwas/data
```

Genotypic, *dominant* and *recessive* tests will not be conducted if any one of the cells in the table of case-control by genotype counts contains less than five observations. This is because the chi-square approximation may not be eliable when cell counts are small. For SNPs with MAFs $< 5\%$, a sample of more than 2,000 cases and controls would be required to meet this threshold and more than 50,000 would be required for SNPs with MAF $< 1\%$.

You can change this default behaviour by adding the flag `--cell`, *e.g.*, we could lower the threshold to 3.

```
plink --bfile gwas/gwa --model --cell 3 --out gwas/data
```

Let's review the contents of the results.

```
gwas_model <- data.table::fread(paste0(COURSE_loc, "/gwas/data.model"))

dim(gwas_model)
```

```
## [1] 1530510      10
```

```
N_SNPS = length(gwas_model$SNP)

gwas_model[1:10, 1:10]
```

```
##     CHR       SNP A1 A2    TEST          AFF       UNAFF   CHISQ DF      P
##  1:   1 rs3934834  T  C    GENO 23/348/1582 23/321/1521 0.26070  2 0.8778
##  2:   1 rs3934834  T  C   TREND    394/3512    367/3363 0.12770  1 0.7209
##  3:   1 rs3934834  T  C ALLELIC    394/3512    367/3363 0.13070  1 0.7177
##  4:   1 rs3934834  T  C     DOM    371/1582    344/1521 0.19060  1 0.6625
##  5:   1 rs3934834  T  C     REC     23/1930     23/1842 0.02475  1 0.8750
##  6:   1 rs3737728  A  G    GENO 206/950/842 222/891/871 2.93100  2 0.2310
##  7:   1 rs3737728  A  G   TREND   1362/2634   1335/2633 0.17780  1 0.6733
##  8:   1 rs3737728  A  G ALLELIC   1362/2634   1335/2633 0.17200  1 0.6783
##  9:   1 rs3737728  A  G     DOM    1156/842    1113/871 1.25700  1 0.2623
## 10:   1 rs3737728  A  G     REC    206/1792    222/1762 0.80220  1 0.3704
```

It contains 1530510 rows, one for each SNP, and each type of test (*genotypic*, *trend*, *allelic*, *dominant*, and *recessive*) and the following columns:

- chromosome [CHR],
- the SNP identifier [SNP],
- the minor allele [A1] (PLINK always codes the A1-allele as the minor allele!),
- the major allele [A2],
- the test performed [TEST]:
  - GENO (genotypic association);
  - TREND (Cochran-Armitage trend);
  - ALLELIC (allelic as- sociation);
  - DOM (dominant model); and
  - REC (recessive model)],
- the cell frequency counts for cases [AFF], and
- the cell frequency counts for controls [UNAFF],
- the chi-square test statistic [CHISQ],
- the degrees of freedom for the test [DF],
- and the asymptotic P value [P] of association.

6.3 Logistic regression

We can also perform a test of association using logistic regression. In this case we might want to correct for covariates/confounding factors, for example age,

sex, ancestral background, i.e. principal components, and other study specific covariates (e.g. hospital of inclusion, genotyping centre etc.). In that case each of these P values is adjusted for the effect of the covariates.

When running a regression analysis, be it linear or logistic, PLINK assumes a multiplicative model. By default, when at least one male and one female is present, sex (male = 1, female = 0) is automatically added as a covariate on X chromosome SNPs, and nowhere else. The `sex` flag causes it to be added everywhere, while `no-x-sex` excludes it.

```
plink --bfile gwas/gwa --logistic sex --covar gwas/gwa.covar --out gwas/data
```

Let's examine the results

```
gwas_assoc <- data.table::fread(paste0(COURSE_loc, "/gwas/data.assoc.logistic"))

dim(gwas_assoc)
```

```
## [1] 918306      9
```

```
gwas_assoc[1:9, 1:9]
```

```
##    CHR       SNP      BP A1 TEST NMISS     OR     STAT      P
## 1:   1 rs3934834  995669  T  ADD  3818 1.0290  0.38120 0.7031
## 2:   1 rs3934834  995669  T  AGE  3818 1.0020  1.11800 0.2635
## 3:   1 rs3934834  995669  T  SEX  3818 1.0120  0.19090 0.8486
## 4:   1 rs3737728 1011278  A  ADD  3982 1.0190  0.38670 0.6990
## 5:   1 rs3737728 1011278  A  AGE  3982 1.0020  1.09800 0.2721
## 6:   1 rs3737728 1011278  A  SEX  3982 1.0060  0.09898 0.9212
## 7:   1 rs6687776 1020428  T  ADD  3915 0.9692 -0.33330 0.7389
## 8:   1 rs6687776 1020428  T  AGE  3915 1.0020  1.04000 0.2984
## 9:   1 rs6687776 1020428  T  SEX  3915 1.0150  0.23690 0.8127
```

If no model option is specified, the first row for each SNP corresponds to results for a multiplicative test of association. The C >= 0 subsequent rows for each SNP correspond to separate tests of significance for each of the C covariates included in the regression model. We can remove the covariate-specific lines from the main report by adding the `hide-covar` flag.

The columns in the association results are: - the chromosome [CHR], - the SNP identifier [SNP], - the base-pair location [BP], - the minor allele [A1], - the test performed [TEST]: ADD (multiplicative model or genotypic model testing additivity), - GENO_2DF (genotypic model), - DOMDEV (genotypic model testing deviation from additivity), - DOM (dominant model), or - REC

(recessive model)], - the number of missing individuals included [NMISS], - the OR relative to the A1, *i.e.* minor allele, - the coefficient z-statistic [STAT], and - the asymptotic P-value [P] of association.

We need to calculate the standard error and confidence interval from the z-statistic. We can modify the effect size (OR) to output the beta by adding the `beta` flag.

6.4 GWAS visualisation

Data visualization is key, not only for presentation but also to inspect the results.

6.4.1 QQ plots

We should create *quantile-quantile (QQ) plots* to compare the observed association test statistics with their expected values under the null hypothesis of no association and so assess the number, magnitude and quality of true associations.

First, we will add the standard error, call rate, A2, and allele frequencies.

```
gwas_assoc_sub <- subset(gwas_assoc, TEST == "ADD")
gwas_assoc_sub$TEST <- NULL

temp <- subset(gwas_FRQ, select = c("SNP", "A2", "MAF", "NCHROBS"))

gwas_assoc_subfrq <- merge(gwas_assoc_sub, temp, by = "SNP")

temp <- subset(gwas_LMISS, select = c("SNP", "callrate"))

gwas_assoc_subfrqlmiss <- merge(gwas_assoc_subfrq, temp, by = "SNP")
head(gwas_assoc_subfrqlmiss)
```

```
##           SNP CHR        BP A1 NMISS     OR    STAT       P A2    MAF NCHROBS
## 1: rs10000010   4  21227772  C  3996 1.0420  0.9010 0.36760  T 0.4258    7992
## 2: rs10000023   4  95952929  T  3957 0.9902 -0.2160 0.82900  G 0.4841    7914
## 3: rs10000030   4 103593179  A  3991 0.9779 -0.3696 0.71170  G 0.1616    7982
## 4:  rs1000007   2 237416793  C  4000 1.0180  0.3649 0.71520  T 0.3122    8000
## 5: rs10000092   4  21504615  C  3963 0.9240 -1.6770 0.09354  T 0.3430    7926
## 6: rs10000121   4 157793485  G  3919 0.9665 -0.7525 0.45170  A 0.4532    7838
##    callrate
## 1:  0.99900
## 2:  0.98925
## 3:  0.99775
```

```
## 4:  1.00000
## 5:  0.99075
## 6:  0.97975
```

```
# Remember:
# - that z = beta/se
# - beta = log(OR), because log is the natural log in r

gwas_assoc_subfrqlmiss$BETA = log(gwas_assoc_subfrqlmiss$OR)
gwas_assoc_subfrqlmiss$SE = gwas_assoc_subfrqlmiss$BETA/gwas_assoc_subfrqlmiss$STAT


gwas_assoc_subfrqlmiss_tib <- dplyr::as_tibble(gwas_assoc_subfrqlmiss)

col_order <- c("SNP", "CHR", "BP",
               "A1", "A2", "MAF", "callrate", "NMISS", "NCHROBS",
               "BETA", "SE", "OR", "STAT", "P")
gwas_assoc_compl <- gwas_assoc_subfrqlmiss_tib[, col_order]

dim(gwas_assoc_compl)
```

```
## [1] 306102     14
```

```
head(gwas_assoc_compl)
```

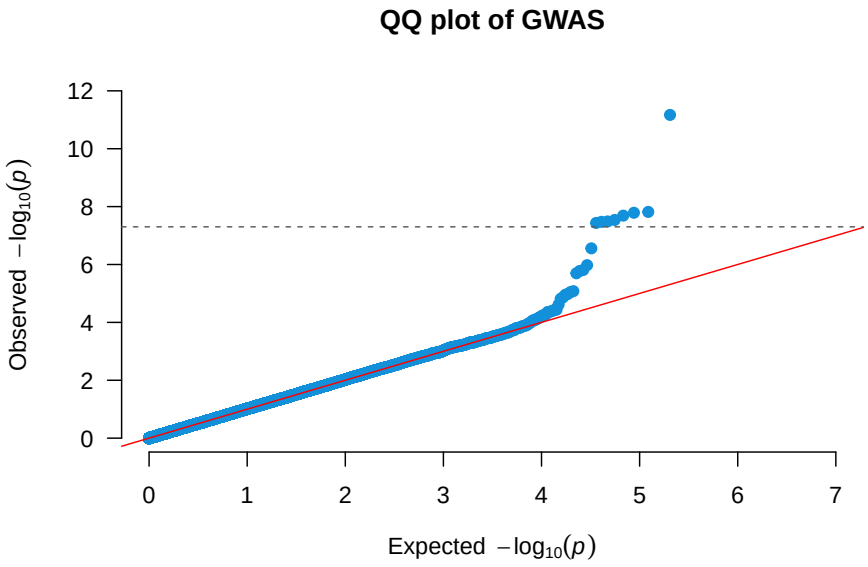
```
## # A tibble: 6 x 14
##   SNP        CHR     BP A1    A2      MAF callrate NMISS NCHROBS     BETA     SE
##   <chr>    <int>  <int> <chr> <chr> <dbl>    <dbl> <int>   <int>    <dbl>  <dbl>
## 1 rs10000~     4 2.12e7 C     T     0.426    0.999  3996    7992  0.0411  0.0457
## 2 rs10000~     4 9.60e7 T     G     0.484    0.989  3957    7914 -0.00985 0.0456
## 3 rs10000~     4 1.04e8 A     G     0.162    0.998  3991    7982 -0.0223  0.0605
## 4 rs10000~     2 2.37e8 C     T     0.312    1      4000    8000  0.0178  0.0489
## 5 rs10000~     4 2.15e7 C     T     0.343    0.991  3963    7926 -0.0790  0.0471
## 6 rs10000~     4 1.58e8 G     A     0.453    0.980  3919    7838 -0.0341  0.0453
## # ... with 3 more variables: OR <dbl>, STAT <dbl>, P <dbl>
```

Let's list the number of SNPs per chromosome.

```
# Number of SNPs per chromosome
knitr::kable(table(gwas_assoc_compl$CHR))
```

Var1	Freq
1	23173
2	25206
3	21402
4	19008
5	19157
6	20672
7	16581
8	18089
9	15709
10	15536
11	14564
12	14889
13	11524
14	9822
15	8838
16	8920
17	8262
18	10356
19	5820
20	7792
21	5412
22	5370

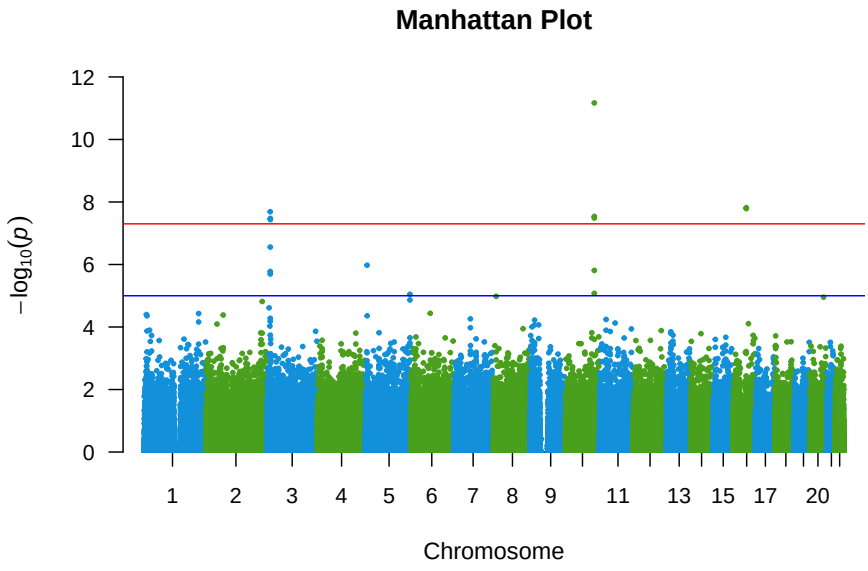
```
gwas_threshold = -log10(5e-8)

qq(gwas_assoc_compl$P, main = "QQ plot of GWAS",
   xlim = c(0, 7),
   ylim = c(0, 12),
   pch = 20, col = uithof_color[16], cex = 1.5, las = 1, bty = "n")
abline(h = gwas_threshold,
       col = uithof_color[25], lty = "dashed")
```



6.4.2 Manhattan plots

We also need to create a *Manhattan plot* to display the association test P-values as a function of chromosomal location and thus provide a visual summary of association test results that draw immediate attention to any regions of significance.

```
manhattan(gwas_assoc_compl, main = "Manhattan Plot",
          ylim = c(0, 12),
          cex = 0.6, cex.axis = 0.9,
          col = c("#1290D9", "#49A01D"))
```



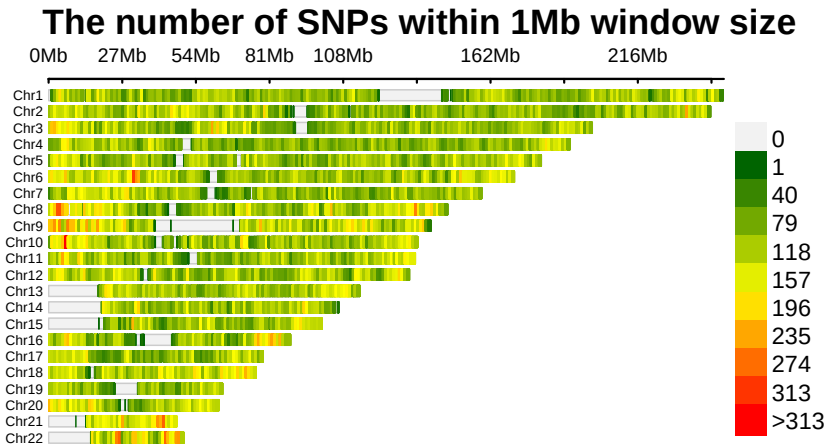

```
gwas_assoc_complsub <- subset(gwas_assoc_compl, select = c("SNP", "CHR", "BP", "P"))
```

6.4.3 Other plots

It is also informative to plot the density per chromosome. We can use the `CMplot` for that which you can find here. For now we just make these graphs 'quick-n-dirty', you can further prettify them, but you easily loose track of time, so maybe carry on.

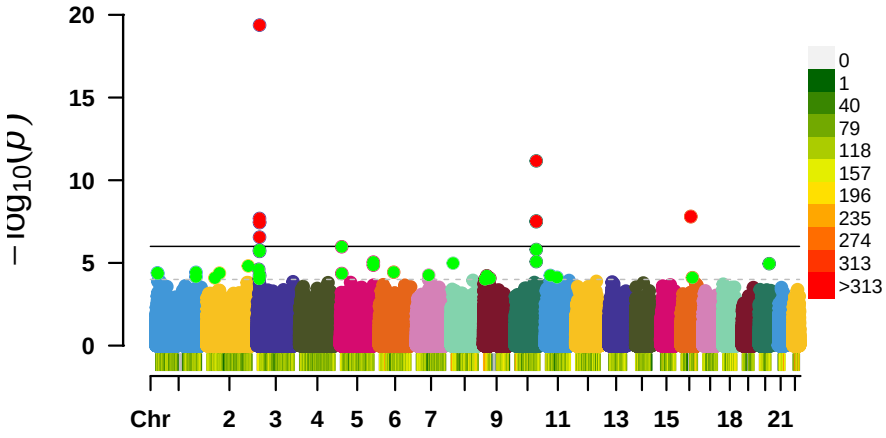
```
CMplot(gwas_assoc_complsub,
       plot.type = "d",
       bin.size = 1e6, col = c("darkgreen", "yellow", "red"),
       file = "jpg", memo = "", dpi = 300, file.output = FALSE, verbose = TRUE)
```

```
##  SNP-Density Plotting.
```



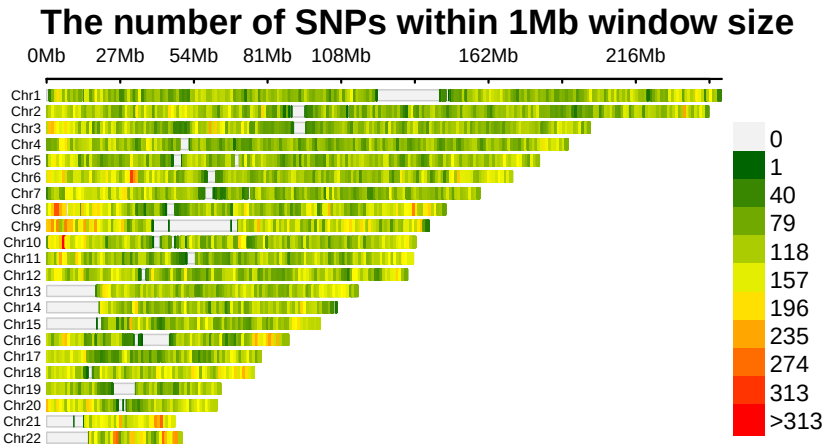

```
CMplot(gwas_assoc_complsub,
       plot.type = "m", LOG10 = TRUE, ylim = NULL,
       threshold = c(1e-6,1e-4), threshold.lty = c(1,2), threshold.lwd = c(1,1), threshold.col =
       amplify = TRUE,
       bin.size = 1e6, chr.den.col = c("darkgreen", "yellow", "red"),
       signal.col = c("red", "green"), signal.cex = c(1,1), signal.pch = c(19,19),
       file = "jpg", memo = "", dpi = 300, file.output = FALSE, verbose = TRUE)
```

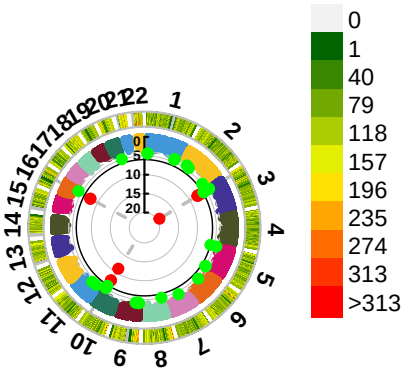
```
##  Rectangular-Manhattan Plotting P.
```



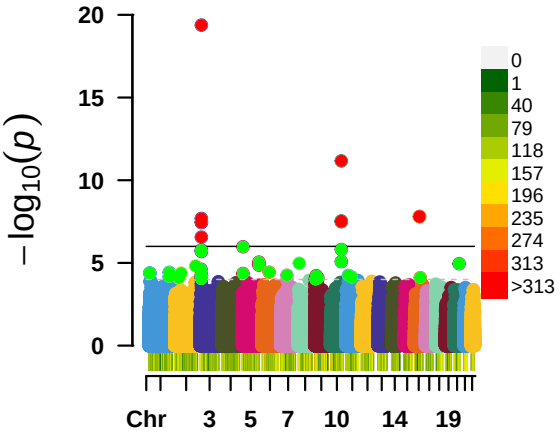

```
CMplot(gwas_assoc_complsub,
       plot.type = "b", LOG10 = TRUE, ylim = NULL,
       threshold = c(1e-6,1e-4), threshold.lty = c(1,2), threshold.lwd = c(1,1), thresh
       amplify = TRUE,
       bin.size = 1e6, chr.den.col = c("darkgreen", "yellow", "red"),
       signal.col = c("red", "green"), signal.cex = c(1,1), signal.pch = c(19,19),
       file = "jpg", memo = "", dpi = 300, file.output = FALSE, verbose = TRUE)
```

```
##  SNP-Density Plotting.
```

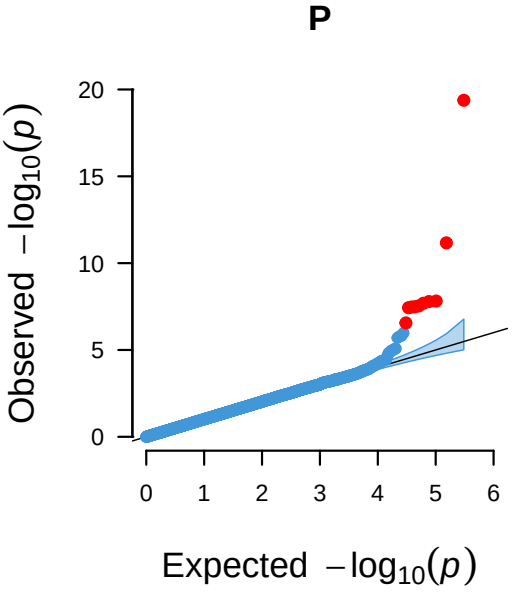





```
##  Circular-Manhattan Plotting P.
##  Rectangular-Manhattan Plotting P.
```




```
##  QQ Plotting P.
```



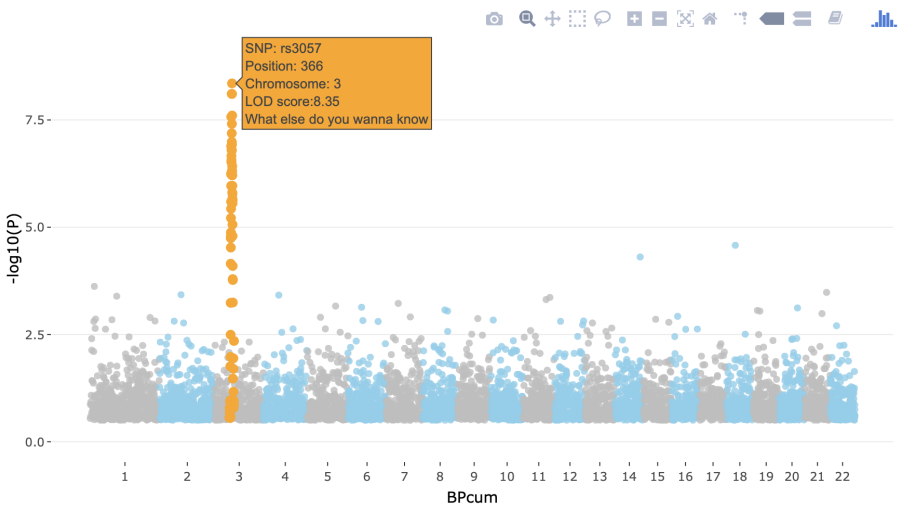
6.4.4 Interactive plots

You can also make an interactive version of the Manhattan - just because you can. The code below shows you how.

```
library(plotly)
library(dplyr)

# Prepare the dataset (as an example we use the data (gwasResults) from the ‘qqman‘-package)
don <- gwasResults %>%

  # Compute chromosome size
  group_by(CHR) %>%
  summarise(chr_len=max(BP)) %>%

  # Calculate cumulative position of each chromosome
  mutate(tot=cumsum(chr_len)-chr_len) %>%
  select(-chr_len) %>%

  # Add this info to the initial dataset
  left_join(gwasResults, ., by=c("CHR"="CHR")) %>%

  # Add a cumulative position of each SNP
  arrange(CHR, BP) %>%
  mutate( BPcum=BP+tot) %>%

  # Add highlight and annotation information
  mutate( is_highlight=ifelse(SNP %in% snpsOfInterest, "yes", "no")) %>%

  # Filter SNP to make the plot lighter
  filter(-log10(P)>0.5)

# Prepare X axis
axisdf <- don %>% group_by(CHR) %>% summarize(center=( max(BPcum) + min(BPcum) ) / 2 )

# Prepare text description for each SNP:
don$text <- paste("SNP: ", don$SNP, "\nPosition: ", don$BP, "\nChromosome: ", don$CHR, "\nLOD sco

# Make the plot
p <- ggplot(don, aes(x=BPcum, y=-log10(P), text=text)) +

    # Show all points
    geom_point( aes(color=as.factor(CHR)), alpha=0.8, size=1.3) +
    scale_color_manual(values = rep(c("grey", "skyblue"), 22 )) +

    # custom X axis:
```

```
    scale_x_continuous( label = axisdf$CHR, breaks= axisdf$center ) +
    scale_y_continuous(expand = c(0, 0), ylim = c(0,9) ) +     # remove space between p

    # Add highlighted points
    geom_point(data=subset(don, is_highlight=="yes"), color="orange", size=2) +

    # Custom the theme:
    theme_bw() +
    theme(
      legend.position="none",
      panel.border = element_blank(),
      panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank()
    )
ggplotly(p, tooltip="text")
```

It will produce something like this.



Again, this is an example with dummy data - you can try to do it for our GWAS, but careful with the time. You can also choose to carry on.

6.4.5 Regional association plots

We can further visualise regions of interest using a package like LocusZoom. But first we need to find the independent hits by *clumping* the results. We will just use the defaults, but please take a note of all the options here https://www.cog-genomics.org/plink/1.9/postproc#clump

```
plink --bfile gwas/gwa --clump gwas/data.assoc.logistic --clump-p1 5e-8 --clump-p2 0.05
```

Now you will have a list of all the *independent* SNPs, *i.e.* the genetic loci, that are associated to the trait.

```
cat gwas/data.assoc.logistic.clumped
```

```
ROOTDIR="/Users/swvanderlaan/Desktop/practical" # change this to your root
cat $ROOTDIR/gwas/data.assoc.logistic.clumped
```

```
##
##  CHR    F         SNP         BP          P    TOTAL   NSIG    S05    S01   S001  S0001
##    3    1   rs6802898   12366207   4.18e-20       50     35      4      2      1      8
##
##                               KB      RSQ  ALLELES    F            P
##   (INDEX)   rs6802898          0    1.000        T    1     4.18e-20
##
##              rs305500       -400   0.0588    TC/CA    1       0.0476
##              rs420014       -394   0.0552    TA/CG    1        0.015
##              rs305494       -392   0.0681    TG/CA    1        0.025
##              rs438129       -383   0.0721    TT/CC    1       0.0126
##             rs7615580       -364    0.309    TC/CT    1     2.05e-08
##              rs307560       -298    0.153    TT/CC    1     1.68e-06
##            rs11720130       -235   0.0838    TA/CG    1      0.00218
##             rs7616006       -124   0.0831    TA/CG    1     5.25e-05
##             rs6775191       -119   0.0506    TG/CA    1     0.000182
##              rs167466       -110   0.0523    TT/CC    1      0.00222
##            rs12635120      -86.6    0.332    TG/CA    1     2.77e-07
##             rs6798713      -85.6    0.288    TC/CT    1     2.01e-06
##             rs2920500      -67.8    0.102    TA/CG    1     6.59e-05
##             rs6768587      -53.1    0.295    TG/CA    1     3.36e-08
##             rs2028760      -18.3    0.305    TA/CG    1     3.67e-08
##
##           RANGE: chr3:11966007..12366207
##            SPAN: 400kb
##
## ----------------------------------------------------------------------
##
##
##  CHR    F         SNP         BP          P    TOTAL   NSIG    S05    S01   S001  S0001
##   10    1   rs7901695  114744078   6.78e-12       32     24      2      1      1      4
##
##                               KB      RSQ  ALLELES    F            P
##   (INDEX)   rs7901695          0    1.000        C    1     6.78e-12
##
##             rs7917983      -21.2   0.0589    CC/TT    1       0.046
##             rs7895307      -10.1   0.0819    CG/TA    1     0.00592
```

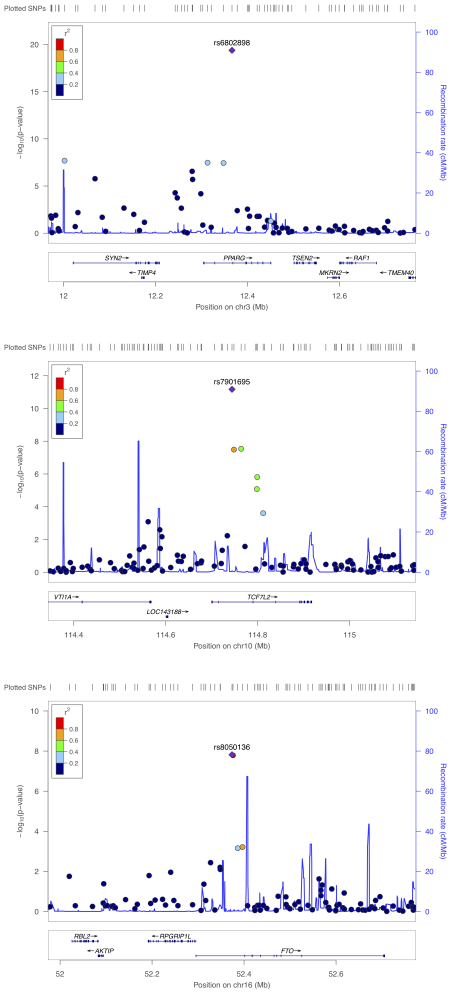
```
##             rs7903146       4.26    0.784    CT/TC    1     3.25e-08
##             rs7904519       19.8    0.582    CG/TA    1     2.89e-08
##            rs11196192       28.2    0.162    CG/TT    1       0.0268
##            rs10885409         54    0.502    CC/TT    1     8.35e-06
##            rs12255372       54.8    0.624    CT/TG    1     1.55e-06
##             rs4918789       67.7     0.24    CG/TT    1     0.000248
##
##           RANGE: chr10:114722872..114811797
##            SPAN: 88kb
##
## ------------------------------------------------------------------
##
##
##  CHR    F         SNP          BP          P    TOTAL   NSIG    S05    S01   S001  S
##   16    1   rs8050136   52373776   1.52e-08       23     16      1      3      2
##
##                              KB      RSQ  ALLELES    F            P
##   (INDEX)   rs8050136          0    1.000        A    1     1.52e-08
##
##             rs7205986      -61.1    0.226    AG/CA    1       0.0434
##             rs6499640      -46.6    0.258    AA/CG    1      0.00367
##             rs1861868      -25.9     0.15    AT/CC    1       0.0063
##             rs1075440      -25.4    0.162    AA/CG    1      0.00802
##             rs3751812       2.18    0.994    AT/CG    1     1.63e-08
##             rs7190492       12.5    0.258    AG/CA    1       0.0007
##             rs8044769       22.9    0.524    AC/CT    1     0.000611
##
##           RANGE: chr16:52312647..52396636
##            SPAN: 83kb
##
## ------------------------------------------------------------------
```

Clumping identifies three loci and now that you know them, you can visualize them using LocusZoom. First, let's get what we need (`SNP` and `P`) and gzip the results.

```
echo "SNP P" > gwas/data.assoc.logistic.locuszoom
cat gwas/data.assoc.logistic | awk '$5=="ADD"' | awk '{ print $2, $9 }' >> gwas/data.as
gzip -v gwas/data.assoc.logistic.locuszoom
```

Now you are ready to upload this `data.assoc.logistic.locuszoom.gz` file to the site: http://locuszoom.org. Try to visualize each locus using the information above and by following the instructions. Choose HapMap 2, hg18, CEU as the LD-reference.

You should get something like below.



You will encounter the above three types of visualizations in any high-quality GWAS paper, because each is so critically informative. Usually, analysts of large-scale meta-analyses of GWAS will also stratify the QQ-plots based on the imputation quality (if your GWAS was imputed), call rate, and allele frequency.

Chapter 7

WTCCC1: a GWAS on coronary artery disease (CAD)

7.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

7.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit

esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

7.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 8

Post-GWAS Analyses

8.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

8.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

8.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 9

Conditional analysis

9.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

9.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

9.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 10

Statistical finemapping

10.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

10.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

10.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 11

Functional Mapping and Annotation of GWAS

11.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

11.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

11.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 12

Phenome-Wide Association Study (PheWAS)

12.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

12.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 13

Mendelian Randomization (MR)

13.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

13.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

13.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 14

Mendelian Randomization (MR)

14.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

14.2 Section 2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

14.3 Section 3

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Chapter 15

License your GitBook

In the spirit of Open Science, it is good to think about making your course materials Open Source. That means that other people can use them. In principle, if you publish materials online without license information, you hold the copyright to those materials. If you want them to be Open Source, you must include a license. It is not always obvious what license to choose.

The Creative Commons licenses are typically suitable for course materials. This GitBook, for example, is licensed under CC-BY 4.0. That means you can use and remix it as you like, but you must credit the original source.

If your project is more focused on software or source code, consider using the GNU GPL v3 license instead.

You can find more information about the Creative Commons Licenses here. Specific licenses that might be useful are:

- CC0 ("No Rights Reserved"), everybody can do what they want with your work.
- CC-BY 4.0 ("Attribution"), everybody can do what they want with your work, but they must credit you. Note that this license may not be suitable for software or source code!

For compatibility between CC and GNU licenses, see this FAQ.

Chapter 16

License your GitBook

In the spirit of Open Science, it is good to think about making your course materials Open Source. That means that other people can use them. In principle, if you publish materials online without license information, you hold the copyright to those materials. If you want them to be Open Source, you must include a license. It is not always obvious what license to choose.

The Creative Commons licenses are typically suitable for course materials. This GitBook, for example, is licensed under CC-BY 4.0. That means you can use and remix it as you like, but you must credit the original source.

If your project is more focused on software or source code, consider using the GNU GPL v3 license instead.

You can find more information about the Creative Commons Licenses here. Specific licenses that might be useful are:

- CC0 ("No Rights Reserved"), everybody can do what they want with your work.
- CC-BY 4.0 ("Attribution"), everybody can do what they want with your work, but they must credit you. Note that this license may not be suitable for software or source code!

For compatibility between CC and GNU licenses, see this FAQ.