

Genetic Association Course

September 26-30, 2022
Max Delbrück Center (MDC) for Molecular Medicine
Berlin, Germany

Computer and Pencil & Paper
Exercises

Table of Contents

Getting Started.....	I
Launching the Exercises.....	II
Introduction to PLINK and R*.....	1
Genome-wide Association Analysis - Quality Control – PLINK**.....	19
Population Genetics & Genetic Epidemiology***.....	30
Linkage Disequilibrium***.....	34
Multifactorial Analysis Controlling for Covariates – PLINK and R*.....	35
Genome-wide Association Analysis - MDS and PCA – PLINK*.....	54
Sequence Data QC and Association Analysis – VAT**.....	60
Association Analysis of Sequence Data using PLINK/SEQ (PSEQ)**.....	78
REGENIE (Jupyter Notebook in JupyterLab).....	N/A
Power and Sample Size Calculations – Cochran Armitage Test for Trend	89
Multiple Test Corrections - PLINK and R*.....	91
Multifactorial Analysis Interactions GXG and GXE – PLINK and R*.....	97

* Can be performed using command line or JupyterLab

**Can be performed using command line or via a Jupyter notebook in JupyterLab

***Pencil and paper exercises

Getting Started

Please view the following videos to install Docker to your computer

For MAC

<https://www.youtube.com/watch?v=DRCDNB1xZ-w>

For Windows PC

<https://www.youtube.com/watch?v=sxv45NCSFMk>

How to install and run course exercises

<https://www.youtube.com/watch?v=OgHvRVtIIog>

For more detail, please read our course wiki

<https://github.com/statgenetics/statgen-courses/wiki/How-to-launch-course-tutorials#alternative-to-cloud-server-use-your-own-computer>

Please go to <https://statgen.us/Tutorials> and install the following tutorials for the course:

Launching Exercises

Please use the following command to launch the exercises. Please note that all exercises (except those that are web based) can be performed either in command line console or JupyterLab environment. Exercises developed in Jupyter Notebook (within a JupyterLab environment) have graphical displays specific to Notebooks and although they can be performed in the command line console if you do so you will not be able to view the graphical output. We recommend you used the JupyterLab environment. Please to not attempt to run Command line and JupyterLab at the same time on your computer, since it will create a file conflict.

- PLINK and R exercises (pages 1-18, 35-53, 91-96, and 97-104)
 - Command line: `statgen-setup login --tutorial plink-r-nothnagel`
 - The command **get-data** may have to be used*
 - JupyterLab: `statgen-setup launch --tutorial plink-r-nothnagel`
- PLINK (pages 19-29 and 54-59)
 - Command line: `statgen-setup login --tutorial plink`
 - The command **get-data** may have to be used*
 - Jupyter Notebook in JupyterLab: `statgen-setup launch --tutorial plink`
- VAT – (pages 60-77)
 - Command line: `statgen-setup login --tutorial vat`
 - The command **get-data** may have to be used*
 - Jupyter Notebook in JupyterLab: `statgen-setup launch --tutorial vat`
- PSEQ (pages 78-88)
 - Command line: `statgen-setup login --tutorial pseq`
 - The command **get-data** may have to be used*
 - Jupyter Notebook in JupyterLab: `statgen-setup launch --tutorial pseq`
- REGENIE – Jupyter Notebook
 - Jupyter Notebook in JupyterLab: `statgen-setup launch --tutorial regenie`
- Power and Sample Size Estimation (pages 89-90)
 - Web-based

*Only necessary to run the **get-data** command if you don't see the data for the exercise already loaded (hint use **ls** command). Before you start the exercise, you will need to `cd` into the work directory (`cd ~/work`).

Exercise

Introduction to PLINK

Running PLINK

PLINK is run at the command line. Additional arguments ('options', 'flags') specify what exactly PLINK should do. All arguments are documented at the PLINK web site (<http://pngu.mgh.harvard.edu/~purcell/plink/>). Under Linux, running PLINK requires to open a shell (or terminal) window. Under Windows, PLINK requires a command prompt ('DOS shell'). Use the shell commands `ls/dir` and `cd` to change the working directory as requested.

When working with PLINK, it is highly recommendable to save all commands in a text file. This way, your work is documented and you will easily (and with certainty) recapitulate what you have done, say, six or twelve month ago. Therefore, also start the text editor and type all commands in some text file, say `PLINK_exercise.q`, and then copy & paste the command lines from the text editor into the shell.

I. The data set

You are provided with a data set on diastolic blood pressure and the genotypes of 20 SNP markers. The data set is already in PLINK format. There are three files:

- **dbp.qt.ped:** Pedigree file with information on family, sex, the quantitative trait (diastolic blood pressure), and genotypes
- **dbp.cc.ped:** Pedigree file with information on family, sex, the dichotomized trait (affected yes/no based on blood pressure), and genotypes
- **dbp.map:** Map file for the SNP markers (*three* columns format)
- **dbp.age.pheno:** Covariate file containing the age of each individual

Use a text editor (notepad/Wordpad under Windows, `pico/vi/nano/emacs` under Linux) to have a look at the contents of these files. Make sure you understand the meaning of each column in the files.

dbp.qt.ped

4928	1	0	0	1	85.51	2	2	1	1	1	...
1838	1	0	0	1	84.51	1	1	1	1	2	...
2450	1	0	0	1	84.3	1	1	1	1	2	...
647	1	0	0	2	89.14	2	2	2	2	1	...
2772	1	0	0	1	90.39	1	2	1	1	1	...
...											

dbp.cc.ped

4928	1	0	0	1	2	2	2	1	1	1	1	0	0	1	...
1838	1	0	0	1	2	1	1	1	1	2	2	2	2	2	...
2450	1	0	0	1	2	1	1	1	1	2	2	2	2	2	...
647	1	0	0	2	2	2	2	2	2	1	2	1	2	2	...
2772	1	0	0	1	2	1	2	1	1	1	2	0	0	1	...
...															

dbp.map

11	rs1101	1021
11	rs1102	3886
11	rs1103	15023
11	rs1104	15788
11	rs1105	21702
...		

dbp.age.pheno

```
4928 1    66
1838 1    67
2450 1    89
647  1    36
2772 1    54
...
```

II. Missing data and filtering

Variables with too many missing values may bias a statistical analysis and lead to spurious results. We will use PLINK to assess the extent of missing values in the data set and to filter variables and samples with too many missing observations.

PLINK requires as the first argument the data set to be processed. This is specified by using the options `--ped` and `--map`. Since the map file contains only three columns instead of the default of four, we additionally have to specify the `--map3` flag. In a first step, we are going to assess the proportion of missing values for each marker and for each sample:

```
plink --ped dbp.cc.ped --map dbp.map --missing
```

Note I: All arguments of a PLINK call have to go on a single line!! Arguments after a line-feed (after pressing the ‘Enter’ key) will be ignored.

Note II: Using a backslash (‘\’) at the end of a line suppressed the line-feed and emulates a continuing line. Using backslashes, a single PLINK call can therefore be distributed over numerous lines. The following PLINK call is identical to the one above:

```
plink --ped dbp.cc.ped \  
      --map dbp.map\  
      --missing
```

PLINK has created three files. The file `plink.log` contains all output from the screen. The files `plink.imiss` and `plink.lmiss` contain the proportion of missing values for each sample and marker, respectively. Use a text editor to have a look at all three files.

plink.log

```
...  
PLINK v1.90b6.9 64-bit (4 Mar 2019)  
Options in effect:  
  --map dbp.map  
  --missing  
  --ped dbp.cc.ped  
...  
Start time: ...  
...  
Scanning .ped file... done.  
Performing single-pass .bed write (20 variants, 600 people).  
--file: plink-temporary.bed + plink-temporary.bim + plink-temporary.fam  
written.  
20 variants loaded from .bim file.  
600 people (329 males, 271 females) loaded from .fam.  
600 phenotype values loaded from .fam.  
Using 1 thread (no multithreaded calculations invoked).  
Before main variant filters, 600 founders and 0 nonfounders present.  
Calculating allele frequencies... done.  
Total genotyping rate is 0.988333.  
--missing: Sample missing data report written to plink.imiss, and variant-based  
missing data report written to plink.lmiss.  
End time: ...
```

plink.imiss

FID	IID	MISS_PHENO	N_MISS	N_GENO	F_MISS
4928	1	N	1	20	0.05
1838	1	N	0	20	0
2450	1	N	1	20	0.05
647	1	N	0	20	0
...					
1284	1	N	2	20	0.1
172	1	N	1	20	0.05
...					

plink.lmiss

CHR	SNP	N_MISS	N_GENO	F_MISS
11	rs1101	0	600	0
11	rs1102	0	600	0
11	rs1103	0	600	0
11	rs1104	92	600	0.1533
11	rs1105	0	600	0
11	rs1106	48	600	0.08
11	rs1107	0	600	0
...				

Next we are going to exclude samples with more than 10% missing genotypes (`--mind 0.10`) and markers with more than 5% (`--geno 0.05`). We will write this filtered data set to a set of new files, called `cleaned.ped` and `cleaned.map`, using `--recode` and `--out`. Further and quality measures and analyses can then be based on this cleaned data set:

```
plink --ped dbp.cc.ped --map dbp.map --mind 0.10 --geno 0.05 \
      --recode --out cleaned
```

PLINK has created three different files. A log file called `cleaned.log` (because we used the `--out` flag) and the two data files `cleaned.ped` and `cleaned.map`. Two markers with too many missing values (rs1104 and rs1106) have been excluded. Use the text editor to have a look at these files. Note that the map file has now the default four columns:

cleaned.map

11	rs1101	0	1021
11	rs1102	0	3886
11	rs1103	0	15023
11	rs1105	0	21702
11	rs1107	0	23508
11	rs1108	0	28769
11	rs1109	0	31385
11	rs1110	0	33198
11	rs1111	0	1245388
11	rs1112	0	1245604
11	rs1113	0	1246723
11	rs1114	0	1246765
11	rs1115	0	1247100
11	rs1116	0	1257557
11	rs1117	0	1258119
11	rs1118	0	1258732
11	rs1119	0	1259178
11	rs1120	0	1259848

We now use this filtered data set to estimate the minor allele frequencies (MAF) of the markers using the `--freq` flag:

```
plink --ped cleaned.ped --map cleaned.map --freq --out cleaned
```

This steps estimates the MAFs, but does *not* filter for a minimum frequency (use the `--maf` flag to this end). The resulting file `cleaned.frq` contains the frequency estimates (check with the text editor). Note that the MAF is always ≤ 0.50 , with the reference allele being *automatically* changed by PLINK! **A2** represents the **major** (more

frequent, ‘baseline’) allele, while **A1** represents the **minor** (less frequent, ‘risk’) allele. *This automatic allele flipping is applied throughout many analyses by PLINK, including association testing!!* You have to carefully check which allele is actually the baseline and which is the minor, or risk, allele in your analysis results! For example, the allele ‘2’ is major (more frequent) allele A2 and allele ‘1’ is the minor (less frequent allele A1 of marker rs1101. In order to avoid automatic allele flipping, use the **--keep-allele-order** flag throughout!

cleaned.frq

CHR	SNP	A1	A2	MAF	NCHROBS
11	rs1101	1	2	0.4508	1200
11	rs1102	2	1	0.2642	1200
11	rs1103	2	1	0.4675	1200
11	rs1105	1	2	0.4558	1200
11	rs1107	2	1	0.1525	1200
11	rs1108	2	1	0.48	1200
...					

Now let’s check for deviations from Hardy-Weinberg equilibrium (HWE):

```
plink --ped cleaned.ped --map cleaned.map --hardy --out cleaned
```

This steps tests for deviations, but does *not* filter for *P*-values below some threshold (use the **--hwe** flag to this end). The resulting file **cleaned.hwe** looks as follows:

cleaned.hwe

CHR	SNP	TEST	A1	A2	GENO	O (HET)	E (HET)	P
11	rs1101	ALL	1	2	115/311/174	0.5183	0.4952	0.2838
11	rs1101	AFF	1	2	54/159/87	0.53	0.4939	0.2426
11	rs1101	UNAFF	1	2	61/152/87	0.5067	0.4962	0.8159
11	rs1102	ALL	2	1	34/249/317	0.415	0.3888	0.115
11	rs1102	AFF	2	1	15/127/158	0.4233	0.3864	0.1339
11	rs1102	UNAFF	2	1	19/122/159	0.4067	0.3911	0.5565
...								

There are three result lines for each marker: one for cases and controls each and one for complete sample. Each line contains the baseline allele (‘A2’), the observed genotype counts (‘GENO’), the observed and expected frequencies of heterozygotes (‘O(HET)’ and ‘E(HET)’), and the corresponding *P*-values (‘P’).

Note: Importing data in PLINK format into R

Importing data in PLINK (LINKAGE) format into R can be sometimes troublesome. A helpful format is the creation of tab-separated text files, where columns are separated by a single, ‘well-defined’ tabular sign (“\t”). However, genotypes are distributed over *two* columns in PLINK format, one for each allele. Since these alleles belong to a single genotype, or variable, a different column separation, e.g. by a space, would be desirable. This can be achieved by additionally using the **--tab** flag:

```
plink --ped cleaned.ped --map cleaned.map --out cleaned.R --recode \
      --tab
```

The resulting text file can be easily read into R using the `read.table` function, used with the argument `sep="\t"`.

A note on larger projects

PLINK can create a large number of files, overwriting existing files without warning. This can be at times confusing. It is also a potent source of errors when it is not clear, say, which filtering criterions actually applied to some particular data set before an analysis. **“Strategic” planning of filtering and exporting steps as well as well-planned naming of files and distributing of output files in different subdirectories is highly recommended with PLINK.**

III. Binary PLINK format

Genome-wide marker genotype data can be massive, resulting in very large file sizes. To reduce file size and speed up calculations, genotype information is usually compressed. Let's convert the text files into binary PLINK format:


```
plink --ped dbp.cc.ped --map dbp.map --make-bed --out dbp
```

PLINK has created four files. The file `dbp.log` contains all output from the screen. File `dbp.fam` contains the family information, `dbp.bim` the marker information and `dbp.bed` the marker genotypes in binary (compressed form). Use a text editor to have a look at the fam and the bim files.

How do these files differ from the previous `dbp.ped` and `dbp.map` files?

Introduction to R

Starting R

For starting R under Windows, simply double-click on the R icon: . This will start the R console where all the commands for R can be entered. Under Linux, R is started by simply typing R at the terminal shell prompt.

When working with R, it is highly recommendable to save all commands in a text file, usually with a `.q`, `.r` or `.R` suffix. This way, your work is documented and you can easily (and with certainty) recapitulate what you have done, say, six or twelve month ago. Therefore, also start a text editor (notepad/Wordpad under Windows, `pico/vi/nano/emacs` under Linux) and type all commands in some text file, say `R_exercise.q`, and then copy & paste command lines from the text editor into the R console.

In many cases, you may also want to change the working directory, i.e. the file folder on your computer where R saves files with exported data and from where it expects to read data files into working memory. Under Windows, this can be done via the menu of the R console. Under Linux, the working directory should be changed at the shell prompt *before* starting R.

If you are unsure how to use a function in R or if you want to specify more arguments of the function, use the help function in R. Simply type `?` and the name of the function at the console, e.g. `?summary`.

I: Data Types

Data can be of different types, for example numeric, strings, or logical values. Suppose we want to compile a (very short) list of European cities with a few features for every city. Enter the following commands (*please* remember to first type these commands into the text editor and only then copy & paste them into the R window):

```
city      = c("Oslo", "Bergen", "Munich", "Berlin", "Rome", "Milan")
population = c(0.58, 0.25, 1.3, 3.4, 2.7, 1.3)
country   = factor( c("Norway" , "Norway", "Germany",
                     "Germany", "Italy", "Italy"  ))
capital   = c(TRUE, FALSE, FALSE, TRUE, TRUE, FALSE)
updated   = 2009
```

You have now created various data objects (city names, population sizes, countries of location, capital status of each cities, year of last update) in the working memory of R by using the assignment operator `'='`. To print the contents of an object, simply type its name:

```
city
population
country
capital
```

Each of these objects is a *vector*, i.e. all elements are of the *same data type*. For example, `city` contains only strings (characters), while `capital` contains only logical values of the cities being the capital of their country or not. Vectors can be concatenated using the `c` function:

```
c(city, city)
c(population, updated)
```

It is often useful to get a short summary of an object. The `summary` function is helpful here:

```
summary (city)
summary (population)
summary (country)
summary (capital)
```

Depending on the data type of an object (or `class` in R), the `summary` function does different things. For example, the mean value can be calculated for numerical variables, but not for nominal ones (represented as `factor` type in R). The type of an object can be assessed by various functions:

```
is.numeric(city)
is.character(city)
is.factor(city)

class (city)
class (population)
class (country)
class (capital)
```

The data type is an attribute of an object. But objects can have more than one attribute. One example is the `length`, which is the number of elements of an object (i.e. number of entries in the vector):

```
length(city)
```

Note: R can also handle objects with elements of *different type* and *length*. The data type `list` is used to represent such data.

II: Names & Indexes

For better data organization, access and presentation, elements in a vector can have names:

```
names(population) = city
population
```

In many data analyses, one would like to access only parts of the complete data set or even only single elements. For example, markers should be tested for deviations from Hardy-Weinberg equilibrium (HWE) separately in affected and unaffected samples. Access to elements of data objects is achieved by means of *indexes*. There are three different kinds of indexes. The simplest one is addressing by *position*:

```
city [3]
city [2:4]
city[c(1,5:6)]
population[3]
```

If the elements of an object have *names*, these names can also be used to access the elements:

```
population["Oslo"]
population[c("Berlin", "Rome")]
```

A third option with vectors is a *logical* index, where only those elements that are marked `TRUE` are accessed. For example, one could select only capitals from the set of all cities:

```
population
capital
population[capital]
```

Logical indexes are quite powerful. One can formulate conditions and store the results in logical vectors. These vectors can then repeatedly be used to access only those elements of the vector that meet the condition. For example, the following commands will select only those cities from our list which have a population of at least one million:

```
population>=1.0  
population[population>=1.0]
```

III: Data frames

Data sets are often presented in a tabular form. Columns usually represent features or measurements and are of the *same* data type. Rows represent observations or samples and may contain possibly *different* data types. For example, work sheets from SPSS or Microsoft Excel as well as tables extracted from SQL databases usually adhere to this format. The corresponding R representation is a *data frame*:

```
cities = data.frame (city=city, pop=population,  
                     country=country, capital=capital, stringsAsFactors=F)  
  
cities  
length(cities)  
dim(cities)  
is.data.frame(cities)  
is.list(cities)
```

Data frames are special lists where all vectors (features) have identical length. They also have some added functionality for printing, summarizing etc. Data frames have two dimensions: rows and columns. Rows (samples) and columns (features) of data frames can also have names:

```
colnames(cities)  
rownames(cities)
```

Indexing is similar to that of vectors. Since there are two dimensions (rows & columns), we need two indexes. We can access single elements as well as complete rows or columns. Logical indexes can also be used:

```
cities$city  
cities[,1]  
cities[2,]  
cities[2,3]  
cities$pop[3]  
cities[capital,]  
cities[cities$pop>=1.0,]
```

IV: Export & Import

All objects in R are held in the *working memory*. After quitting R, all objects are lost unless they have been saved in external files on the computer disk!! There are several possibilities to save objects to the disk.

First, let's have an overview on which objects are currently held in the working memory:

```
ls()
```

Now save the objects `cities`, `city`, and `country` in an external archive file called `myobjects.R`. Note that this file is in a format that is only readable with R!

```
save(cities, city, country, file="myobjects.R")
```

It is often useful to export your data set into text format, so that the data can be read with a text editor, such as Word, or be imported into other software programs. For data frames, this can be done with the `write.table` function:

```
write.table(cities, file="cities.txt")
```

Output can also be re-directed from the R console to some text file:

```
sink ("cities.output.txt")  
print (cities)
```

```
sink ()
```

Now check if these files have properly been created in the working directory of your computer:

```
dir()
```

This command lists the contents of the current working directory on your computer hard disk. If the files `cities.txt` and `cities.output.txt` have been not been created by R, check for possible errors in your script or ask for help. If these files have been created, delete all objects from the R working memory:

```
rm(list=ls())  
ls()
```

No objects are currently held in the working memory. Import the data frame from the external text file `cities.txt` using the `read.table` function and assign it to some object called `new.table`:

```
new.table = read.table ("cities.txt")  
ls()  
new.table
```

Next, import the objects from the R archive file `myobjects.R` using the `load` function:

```
load ("myobjects.R")  
ls()  
cities  
new.table
```

Quitting R

Quit the R session by entering the following command and answer no to the upcoming question:

```
q()
```

Answers

Introduction to PLINK

I: The data set

dbp.qt.ped

4928	1	0	0	1	85.51	2	2	1	1	1	...
1838	1	0	0	1	84.51	1	1	1	1	2	...
2450	1	0	0	1	84.3	1	1	1	1	2	...
647	1	0	0	2	89.14	2	2	2	2	1	...
2772	1	0	0	1	90.39	1	2	1	1	1	...
...											

dbp.cc.ped

4928	1	0	0	1	2	2	2	1	1	1	1	0	0	1	...
1838	1	0	0	1	2	1	1	1	1	2	2	2	2	2	...
2450	1	0	0	1	2	1	1	1	1	2	2	2	2	2	...
647	1	0	0	2	2	2	2	2	2	1	2	1	2	2	...
2772	1	0	0	1	2	1	2	1	1	1	2	0	0	1	...
...															

dbp.map

11	rs1101	1021
11	rs1102	3886
11	rs1103	15023
11	rs1104	15788
11	rs1105	21702
...		

dbp.age.pheno

4928	1	66
1838	1	67
2450	1	89
647	1	36
2772	1	54
...		

II. Missing data and filtering

```
plink --ped dbp.cc.ped --map dbp.map --missing
```

```
PLINK v1.90b6.9 64-bit (4 Mar 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang  GNU General Public License v3
Logging to plink.log.
Options in effect:
  --map dbp.map
  --missing
  --ped dbp.cc.ped
```

```
16384 MB RAM detected; reserving 8192 MB for main workspace.
.ped scan complete (for binary autoconversion).
Performing single-pass .bed write (20 variants, 600 people).
--file: plink-temporary.bed + plink-temporary.bim + plink-temporary.fam
written.
20 variants loaded from .bim file.
600 people (329 males, 271 females) loaded from .fam.
600 phenotype values loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 600 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.988333.
--missing: Sample missing data report written to plink.imiss, and variant-based missing
data report written to plink.lmiss.
```

The screen printout documented above is also contained in the file `plink.log`. PLINK has also generated two other files. The files `plink.imiss` and `plink.lmiss` contain the proportion of missing values for each sample and marker, respectively.

plink.imiss

FID	IID	MISS_PHENO	N_MISS	N_GENO	F_MISS
4928	1	N	1	20	0.05
1838	1	N	0	20	0
2450	1	N	1	20	0.05
647	1	N	0	20	0
...					
1284	1	N	2	20	0.1
172	1	N	1	20	0.05
...					

plink.lmiss

CHR	SNP	N_MISS	N_GENO	F_MISS
11	rs1101	0	600	0
11	rs1102	0	600	0
11	rs1103	0	600	0
11	rs1104	92	600	0.1533
11	rs1105	0	600	0
11	rs1106	48	600	0.08
11	rs1107	0	600	0
11	rs1108	0	600	0
11	rs1109	0	600	0
11	rs1110	0	600	0

11	rs1111	0	600	0
11	rs1112	0	600	0
11	rs1113	0	600	0
11	rs1114	0	600	0
11	rs1115	0	600	0
11	rs1116	0	600	0
11	rs1117	0	600	0
11	rs1118	0	600	0
11	rs1119	0	600	0
11	rs1120	0	600	0

```
plink --ped dbp.cc.ped --map dbp.map --mind 0.10 --geno 0.05 \
      --recode --out cleaned
```

```
PLINK v1.90b6.9 64-bit (4 Mar 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang GNU General Public License v3
Logging to cleaned.log.
```

Options in effect:

```
--geno 0.05
--map dbp.map
--mind 0.10
--out cleaned
--ped dbp.cc.ped
--recode
```

```
16384 MB RAM detected; reserving 8192 MB for main workspace.
.ped scan complete (for binary autoconversion).
Performing single-pass .bed write (20 variants, 600 people).
--file: cleaned-temporary.bed + cleaned-temporary.bim + cleaned-temporary.fam
written.
20 variants loaded from .bim file.
600 people (329 males, 271 females) loaded from .fam.
600 phenotype values loaded from .fam.
0 people removed due to missing genotype data (--mind).
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 600 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.988333.
2 variants removed due to missing genotype data (--geno).
18 variants and 600 people pass filters and QC.
Among remaining phenotypes, 300 are cases and 300 are controls.
--recode ped to cleaned.ped + cleaned.map ... done.
```

PLINK has created three different files. A log file called `cleaned.log` (because we used the `--out` flag) and the two data files `cleaned.map` and `cleaned.ped`. Two markers with too many missing values (rs1104 and rs1106) have been excluded. Use the text editor to have a look at these files. Note that the map file has now the default four columns:

cleaned.map

11	rs1101	0	1021
11	rs1102	0	3886
11	rs1103	0	15023
11	rs1105	0	21702
11	rs1107	0	23508
11	rs1108	0	28769
11	rs1109	0	31385
11	rs1110	0	33198
11	rs1111	0	1245388
11	rs1112	0	1245604
11	rs1113	0	1246723

11	rs1114	0	1246765
11	rs1115	0	1247100
11	rs1116	0	1257557
11	rs1117	0	1258119
11	rs1118	0	1258732
11	rs1119	0	1259178
11	rs1120	0	1259848

cleaned.ped

4928	1	0	0	1	2	2	2	1	1	1	1	1	1	1	2	2	1	1	1	1	2	1	2	1	2	1	2	2	2	1	1	...			
1838	1	0	0	1	2	1	1	1	1	2	2	2	1	1	1	2	2	2	2	2	1	1	1	1	2	2	2	2	2	2	1	1	...		
2450	1	0	0	1	2	1	1	1	1	2	2	2	2	1	1	1	2	2	2	2	2	2	1	2	1	2	2	1	2	2	1	1	...		
647	1	0	0	2	2	2	2	2	2	2	1	2	2	2	1	2	1	2	2	1	2	1	2	1	1	1	1	2	2	2	2	1	...		
2772	1	0	0	1	2	1	2	1	1	2	1	1	2	1	1	2	1	1	2	1	2	1	1	1	1	1	1	1	2	2	2	1	1	...	
148	1	0	0	2	2	2	2	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	1	2	1	1	1	1	2	2	2	1	...	
1	1	0	0	1	2	1	2	2	1	2	2	2	2	1	1	1	2	2	1	2	1	1	1	1	2	1	2	1	2	2	1	2	1	1	...
...																																			

plink --ped cleaned.ped --map cleaned.map --freq --out cleaned

PLINK v1.90b6.9 64-bit (4 Mar 2019) www.cog-genomics.org/plink/1.9/
 (C) 2005-2019 Shaun Purcell, Christopher Chang GNU General Public License v3
 Logging to cleaned.log.
 Options in effect:
 --freq
 --map cleaned.map
 --out cleaned
 --ped cleaned.ped

16384 MB RAM detected; reserving 8192 MB for main workspace.
 .ped scan complete (for binary autoconversion).
 Performing single-pass .bed write (18 variants, 600 people).
 --file: cleaned-temporary.bed + cleaned-temporary.bim + cleaned-temporary.fam
 written.
 18 variants loaded from .bim file.
 600 people (329 males, 271 females) loaded from .fam.
 600 phenotype values loaded from .fam.
 Using 1 thread (no multithreaded calculations invoked).
 Before main variant filters, 600 founders and 0 nonfounders present.
 Calculating allele frequencies... done.
 --freq: Allele frequencies (founders only) written to cleaned.frq

PLINK has created the file cleaned.frq, containing the frequency estimates. The log file cleaned.log has been overwritten:

cleaned.frq

CHR	SNP	A1	A2	MAF	NCHROBS
11	rs1101	1	2	0.4508	1200
11	rs1102	2	1	0.2642	1200
11	rs1103	2	1	0.4675	1200
11	rs1105	1	2	0.4558	1200
11	rs1107	2	1	0.1525	1200
11	rs1108	2	1	0.48	1200
11	rs1109	1	2	0.4425	1200
11	rs1110	1	2	0.4558	1200
11	rs1111	2	1	0.435	1200
11	rs1112	2	1	0.2958	1200
11	rs1113	2	1	0.2683	1200
11	rs1114	2	1	0.4175	1200
11	rs1115	1	2	0.2642	1200

11	rs1116	1	2	0.08	1200
11	rs1117	2	1	0.1817	1200
11	rs1118	2	1	0.2842	1200
11	rs1119	1	2	0.185	1200
11	rs1120	1	2	0.3025	1200

plink --ped cleaned.ped --map cleaned.map --hardy --out cleaned

PLINK v1.90b6.9 64-bit (4 Mar 2019) www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang GNU General Public License v3
Logging to cleaned.log.

Options in effect:

--hardy
--map cleaned.map
--out cleaned
--ped cleaned.ped

16384 MB RAM detected; reserving 8192 MB for main workspace.

.ped scan complete (for binary autoconversion).

Performing single-pass .bed write (18 variants, 600 people).

--file: cleaned-temporary.bed + cleaned-temporary.bim + cleaned-temporary.fam
written.

18 variants loaded from .bim file.

600 people (329 males, 271 females) loaded from .fam.

600 phenotype values loaded from .fam.

Using 1 thread (no multithreaded calculations invoked).

Before main variant filters, 600 founders and 0 nonfounders present.

Calculating allele frequencies... done.

--hardy: Writing Hardy-Weinberg report (founders only) to cleaned.hwe ... done.

PLINK has created the file cleaned.hwe, containing the *P* values for the test of deviation from Hardy-Weinberg equilibrium (HWE). The log file cleaned.log has again been overwritten:

cleaned.hwe

CHR	SNP	TEST	A1	A2	GENO	O (HET)	E (HET)	P
11	rs1101	ALL	1	2	115/311/174	0.5183	0.4952	0.2838
11	rs1101	AFF	1	2	54/159/87	0.53	0.494	0.2426
11	rs1101	UNAFF	1	2	61/152/87	0.5067	0.4962	0.8159
11	rs1102	ALL	2	1	34/249/317	0.415	0.3888	0.115
11	rs1102	AFF	2	1	15/127/158	0.4233	0.3864	0.1339
11	rs1102	UNAFF	2	1	19/122/159	0.4067	0.3911	0.5565
11	rs1103	ALL	2	1	126/309/165	0.515	0.4979	0.4136
11	rs1103	AFF	2	1	57/159/84	0.53	0.496	0.2943
11	rs1103	UNAFF	2	1	69/150/81	0.5	0.4992	1
11	rs1105	ALL	1	2	118/311/171	0.5183	0.4961	0.2859
11	rs1105	AFF	1	2	56/165/79	0.55	0.4971	0.08129
11	rs1105	UNAFF	1	2	62/146/92	0.4867	0.495	0.8155
11	rs1107	ALL	2	1	13/157/430	0.2617	0.2585	0.8749
11	rs1107	AFF	2	1	6/85/209	0.2833	0.2711	0.5274
11	rs1107	UNAFF	2	1	7/72/221	0.24	0.2456	0.6406
11	rs1108	ALL	2	1	139/298/163	0.4967	0.4992	0.9348
11	rs1108	AFF	2	1	74/152/74	0.5067	0.5	0.9081
11	rs1108	UNAFF	2	1	65/146/89	0.4867	0.4968	0.7281
11	rs1109	ALL	1	2	113/305/182	0.5083	0.4934	0.508
11	rs1109	AFF	1	2	56/154/90	0.5133	0.4936	0.5587
11	rs1109	UNAFF	1	2	57/151/92	0.5033	0.4932	0.8148
11	rs1110	ALL	1	2	112/323/165	0.5383	0.4961	0.04003
11	rs1110	AFF	1	2	50/169/81	0.5633	0.4947	0.01965
11	rs1110	UNAFF	1	2	62/154/84	0.5133	0.4973	0.6426
11	rs1111	ALL	2	1	116/290/194	0.4833	0.4915	0.6785
11	rs1111	AFF	2	1	62/140/98	0.4667	0.4928	0.3509
11	rs1111	UNAFF	2	1	54/150/96	0.5	0.4902	0.8138

11	rs1112	ALL	2	1	52/251/297	0.4183	0.4166	1
11	rs1112	AFF	2	1	39/145/116	0.4833	0.4671	0.6212
11	rs1112	UNAFF	2	1	13/106/181	0.3533	0.3432	0.7367
11	rs1113	ALL	2	1	43/236/321	0.3933	0.3927	1
11	rs1113	AFF	2	1	27/136/137	0.4533	0.4328	0.5044
11	rs1113	UNAFF	2	1	16/100/184	0.3333	0.3432	0.6146
11	rs1114	ALL	2	1	111/279/210	0.465	0.4864	0.2764
11	rs1114	AFF	2	1	52/137/111	0.4567	0.4807	0.401
11	rs1114	UNAFF	2	1	59/142/99	0.4733	0.4911	0.5568
11	rs1115	ALL	1	2	45/227/328	0.3783	0.3888	0.5286
11	rs1115	AFF	1	2	35/127/138	0.4233	0.4411	0.5128
11	rs1115	UNAFF	1	2	10/100/190	0.3333	0.32	0.5887
11	rs1116	ALL	1	2	4/88/508	0.1467	0.1472	0.785
11	rs1116	AFF	1	2	3/43/254	0.1433	0.15	0.4294
11	rs1116	UNAFF	1	2	1/45/254	0.15	0.1444	1
11	rs1117	ALL	2	1	14/190/396	0.3167	0.2973	0.1309
11	rs1117	AFF	2	1	12/117/171	0.39	0.3595	0.1974
11	rs1117	UNAFF	2	1	2/73/225	0.2433	0.2237	0.1935
...								

PLINK has performed HWE tests for each marker in each of three sample sets: controls ('UNAFF'), cases ('AFF') and controls and cases combined ('ALL'). The 'GENO' column gives the counts of A1/A1, A1/A2 and A2/A2 genotypes in the sample set, respectively. The columns 'O (HET)' and 'E (HET)' give the observed and the expected frequency of heterozygous genotypes A1/A2 according to the Hardy-Weinberg proportions (i.e. $2pq$ if p denotes the frequency of the A1 allele and q that of the A2 allele). The 'P' column contains the P -value from the test.

```
plink --ped cleaned.ped --map cleaned.map --out cleaned.R --recode \
      --tab
```

```
PLINK v1.90b6.9 64-bit (4 Mar 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang  GNU General Public License v3
Logging to cleaned.R.log.
```

```
Options in effect:
```

```
--map cleaned.map
--out cleaned.R
--ped cleaned.ped
--recode
--tab
```

```
Note: --tab flag deprecated. Use '--recode tab ...'.
```

```
16384 MB RAM detected; reserving 8192 MB for main workspace.
```

```
.ped scan complete (for binary autoconversion).
```

```
Performing single-pass .bed write (18 variants, 600 people).
```

```
--file: cleaned.R-temporary.bed + cleaned.R-temporary.bim +
cleaned.R-temporary.fam written.
```

```
18 variants loaded from .bim file.
```

```
600 people (329 males, 271 females) loaded from .fam.
```

```
600 phenotype values loaded from .fam.
```

```
Using 1 thread (no multithreaded calculations invoked).
```

```
Before main variant filters, 600 founders and 0 nonfounders present.
```

```
Calculating allele frequencies... done.
```

```
18 variants and 600 people pass filters and QC.
```

```
Among remaining phenotypes, 300 are cases and 300 are controls.
```

```
--recode ped to cleaned.R.ped + cleaned.R.map ... done.
```

cleaned.R.ped

4928	1	0	0	1	2	2	2	1	1	1	1	1	1	..
1838	1	0	0	1	2	1	1	1	1	2	2	2	2	..
2450	1	0	0	1	2	1	1	1	1	2	2	2	2	..
647	1	0	0	2	2	2	2	2	2	2	1	2	2	..
2772	1	0	0	1	2	1	2	1	1	2	1	1	2	..
148	1	0	0	2	2	2	2	1	1	1	1	1	1	..
1	1	0	0	1	2	1	2	2	1	2	2	2	2	..
1696	1	0	0	2	2	1	2	2	1	2	1	1	2	..
890	1	0	0	1	2	1	2	1	1	2	1	1	2	..
1832	1	0	0	1	2	1	2	1	1	2	1	1	2	..
...														

cleaned.R.map

11	rs1101	0	1021
11	rs1102	0	3886
11	rs1103	0	15023
11	rs1105	0	21702
11	rs1107	0	23508
11	rs1108	0	28769
11	rs1109	0	31385
11	rs1110	0	33198
...			

II. Missing data and filtering

plink --ped dbp.cc.ped --map dbp.map --missing

PLINK v1.90b4.4 64-bit (21 May 2017) www.cog-genomics.org/plink/1.9/
(C) 2005-2017 Shaun Purcell, Christopher Chang GNU General Public License v3

Logging to dbp.log.

Options in effect:

- make-bed
- map dbp.map
- out dbp
- ped dbp.cc.ped

16384 MB RAM detected; reserving 8192 MB for main workspace.

.ped scan complete (for binary autoconversion).

Performing single-pass .bed write (20 variants, 600 people).

--file: dbp-temporary.bed + dbp-temporary.bim + dbp-temporary.fam written.

20 variants loaded from .bim file.

600 people (329 males, 271 females) loaded from .fam.

600 phenotype values loaded from .fam.

Using 1 thread (no multithreaded calculations invoked).

Before main variant filters, 600 founders and 0 nonfounders present.

Calculating allele frequencies... done.

Total genotyping rate is 0.988333.

20 variants and 600 people pass filters and QC.

Among remaining phenotypes, 300 are cases and 300 are controls.

--make-bed to dbp.bed + dbp.bim + dbp.fam ... done.

dbp.fam

```
4928 1 0 0 1 2
1838 1 0 0 1 2
2450 1 0 0 1 2
647 1 0 0 2 2
2772 1 0 0 1 2
148 1 0 0 2 2
1 1 0 0 1 2
1696 1 0 0 2 2
890 1 0 0 1 2
1832 1 0 0 1 2
...
```

dbp.bim

```
11 rs1101 0 1021 1 2
11 rs1102 0 3886 2 1
11 rs1103 0 15023 2 1
11 rs1104 0 15788 1 2
11 rs1105 0 21702 1 2
11 rs1106 0 23505 2 1
11 rs1107 0 23508 2 1
11 rs1108 0 28769 2 1
11 rs1109 0 31385 1 2
11 rs1110 0 33198 1 2
...
```

File `dbp.fam` contains the first six columns of `dbp.ped`, whereas `dbp.bim` contains all four columns from `dbp.map` and two additional columns from the `dbp.ped` file listing the A2 and A1 alleles.

Introduction to R

I: Data Types

```
city      = c("Oslo", "Bergen", "Munich", "Berlin", "Rome", "Milan")
population = c(0.58, 0.25, 1.3, 3.4, 2.7, 1.3)
country   = factor ( c("Norway", "Norway", "Germany",
                      "Germany", "Italy", "Italy") )
capital   = c(TRUE, FALSE, FALSE, TRUE, TRUE, FALSE)
updated   = 2009
city
[1] "Oslo"   "Bergen" "Munich" "Berlin" "Rome"   "Milan"
population
[1] 0.58 0.25 1.30 3.40 2.70 1.30
country
[1] Norway Norway Germany Germany Italy Italy
Levels: Germany Italy Norway
capital
[1] TRUE FALSE FALSE TRUE TRUE FALSE

c(city, city)
[1] "Oslo"   "Bergen" "Munich" "Berlin" "Rome"   "Milan" "Oslo"   "Bergen" "Munich"
"Berlin" "Rome"   "Milan"
c(population, updated)
[1] 0.58 0.25 1.30 3.40 2.70 1.30 2009.00
```

```

summary (city)
Length      Class      Mode
      6 character character
summary (population)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.250   0.760   1.300   1.588   2.350   3.400
summary (country)
Germany  Italy  Norway
      2      2      2
summary (capital)
      Mode  FALSE    TRUE    NA's
logical    3      3      0

is.numeric(city)
[1] FALSE
is.character(city)
[1] TRUE
is.factor(city)
[1] FALSE
class (city)
[1] "character"
class (population)
[1] "numeric"
class (country)
[1] "factor"
class (capital)
[1] "logical"
length(city)
[1] 6

```

II: Names & Indexes

```

names(population) = city
  Oslo Bergen Munich Berlin  Rome  Milan
  0.58  0.25  1.30  3.40  2.70  1.30

population
city [3]
[1] "Munich"
city [2:4]
[1] "Bergen" "Munich" "Berlin"
city[c(1,5:6)]
[1] "Oslo" "Rome" "Milan"
population[3]
Munich
  1.3
population["Oslo"]
Oslo
  0.58
population[c("Berlin", "Rome")]
Berlin  Rome
  3.4    2.7
population[capital]
  Oslo Berlin  Rome
  0.58  3.40  2.70
population>=1.0
  Oslo Bergen Munich Berlin  Rome  Milan
FALSE FALSE  TRUE  TRUE  TRUE  TRUE
population[population>=1.0]
Munich Berlin  Rome  Milan
  1.3    3.4    2.7    1.3

```

III: Data frames

```
cities = data.frame (city=city, pop=population,
                     country=country, capital=capital,
                     stringsAsFactors = F)
```

```
cities
```

```
      city pop country capital
Oslo    Oslo 0.58  Norway    TRUE
Bergen Bergen 0.25  Norway    FALSE
Munich Munich 1.30 Germany    FALSE
Berlin Berlin 3.40 Germany    TRUE
Rome     Rome 2.70   Italy     TRUE
Milan    Milan 1.30   Italy     FALSE
```

```
length(cities)
```

```
[1] 4
```

```
dim(cities)
```

```
[1] 6 4
```

```
is.data.frame(cities)
```

```
[1] TRUE
```

```
is.list(cities)
```

```
[1] TRUE
```

```
colnames(cities)
```

```
[1] "city"      "pop"        "country" "capital"
```

```
rownames(cities)
```

```
[1] "Oslo"      "Bergen" "Munich" "Berlin" "Rome"    "Milan"
```

```
cities$city
```

```
[1] "Oslo"      "Bergen" "Munich" "Berlin" "Rome"    "Milan"
```

```
cities[,1]
```

```
[1] "Oslo"      "Bergen" "Munich" "Berlin" "Rome"    "Milan"
```

```
cities[2,]
```

```
      city pop country capital
Bergen Bergen 0.25  Norway    FALSE
```

```
cities[2,3]
```

```
[1] Norway
```

```
Levels: Germany Italy Norway
```

```
cities$pop[3]
```

```
[1] 1.3
```

```
cities[capital,]
```

```
      city pop country capital
Oslo    Oslo 0.58  Norway    TRUE
Berlin Berlin 3.40 Germany    TRUE
Rome     Rome 2.70   Italy     TRUE
```

```
cities[cities$pop>=1.0,]
```

```
      city pop country capital
Munich Munich 1.3 Germany    FALSE
Berlin Berlin 3.4 Germany    TRUE
Rome     Rome 2.7   Italy     TRUE
Milan    Milan 1.3   Italy     FALSE
```

IV: Export & Import

```
ls()
```

```
[1] "capital"      "cities"        "city"          "country"
```

```
[5] "population"   "updated"
```

```
save(cities, city, country, file="myobjects.R")
```

```

write.table(cities, file="cities.txt")

sink ("cities.output.txt")
print (cities)
sink ()

dir()
[1] "R_exercise.txt"  "cities.output.txt"  "myobjects.R"

rm(list=ls())
ls()
character(0)

new.table = read.table ("cities.txt")
ls()
[1] "new.table"
new.table
      city pop country capital
Oslo    Oslo 0.58  Norway    TRUE
Bergen Bergen 0.25  Norway    FALSE
Munich  Munich 1.30 Germany    FALSE
Berlin  Berlin 3.40 Germany    TRUE
Rome     Rome 2.70   Italy     TRUE
Milan    Milan 1.30   Italy     FALSE

load ("myobjects.R")
ls()
[1] "cities"      "city"        "country"     "new.table"

cities
      city pop country capital
Oslo    Oslo 0.58  Norway    TRUE
Bergen Bergen 0.25  Norway    FALSE
Munich  Munich 1.30 Germany    FALSE
Berlin  Berlin 3.40 Germany    TRUE
Rome     Rome 2.70   Italy     TRUE
Milan    Milan 1.30   Italy     FALSE

new.table
      city pop country capital
Oslo    Oslo 0.58  Norway    TRUE
Bergen Bergen 0.25  Norway    FALSE
Munich  Munich 1.30 Germany    FALSE
Berlin  Berlin 3.40 Germany    TRUE
Rome     Rome 2.70   Italy     TRUE
Milan    Milan 1.30   Italy     FALSE

```

Genome-wide Association Analysis - Data Quality Control

Copyright © 2022 Merry-Lynn McDonald, Isabelle Schrauwen & Suzanne M. Leal

Introduction

In this exercise, you will learn how to perform data quality control (QC) by removing markers and samples that fail QC quality control criteria. You will also examine your samples for individuals that are related to each other and/or are duplicate samples. Each sample will also be tested for excess homozygosity and heterozygosity of genotype data. Each SNP will be tested for deviations from Hardy-Weinberg Equilibrium. These exercises will be carried out using PLINK1.9 and R.

1. Using PLINK

PLINK can upload data in different formats please see the PLINK documentation (<https://www.cog-genomics.org/plink/1.9/input>) for additional details. The data for this exercise is in PLINK/LINKAGE file format. There are two files: a pedfile (GWAS.ped) and a map file (GWAS.map). Please examine these files and the PLINK documentation. Please note the commands must be given in the directory where the data resides.

Navigate via the command prompt to the directory which contains the files for the exercise. Type **plink** in the command prompt and make note of the output. Next type:

```
plink --file GWAS
```

Note, that PLINK outputs a file called **plink.log** that contains the same output which you see on the screen. To see all options, type `plink --help` for more information. Determine how many samples there are in your data set and fill in Oval 1 of the flowchart below.

2. Data Quality Control

a. Removing Samples and SNPs with Missing Genotypes.

You will exclude samples that are missing more than 10% of their genotype calls. These samples are likely to have been generated using low quality DNA and can also have higher than average genotyping error rates.

```
plink --file GWAS --mind 0.10 --recode --out GWAS_clean_mind
```

Examine **GWAS_clean_mind.log** to see how many samples are excluded based on this criterion and fill in Box 1.

Create two versions of your dataset, one with SNPs with a minor allele frequencies (MAFs) $\geq 5\%$ and the other with SNPs with a MAFs $< 5\%$.

You will now remove SNPs with MAFs $\geq 5\%$ that are missing $> 5\%$ of their genotypes and then remove SNPs with MAFs $< 5\%$ that are missing $> 1\%$ of their genotypes. SNPs which are missing genotypes can have higher error rates than those SNP markers without missing data.

```
plink --file GWAS_clean_mind --maf 0.05 --recode --out MAF_greater_5
plink --file GWAS_clean_mind --exclude MAF_greater_5.map --recode --out MAF_less_5

plink --file MAF_greater_5 --geno 0.05 --recode --out MAF_greater_5_clean
```

Fill in Box 2a.

```
plink --file MAF_less_5 --geno 0.01 --recode --out MAF_less_5_clean
```

Fill in Box 2b.

Merge the two files.

```
plink --file MAF_greater_5_clean --merge MAF_less_5_clean.ped MAF_less_5_clean.map  
--recode --out GWAS_MAF_clean
```

A more stringent criterion for missing data is used, samples missing >3% of their genotypes are removed.

```
plink --file GWAS_MAF_clean --mind 0.03 --recode --out GWAS_clean2
```

Fill in Box 3.

b. Checking Sex

Error of the reported sex of an individual can occur. Information from the SNP genotypes can be used to verify the sex of individuals, by examining homozygosity (F) on the X chromosome for every individual. F is expected to be <0.2 in females and >0.8 in males. To check sex run

```
plink --file GWAS_clean2 --check-sex --out GWAS_sex_checking
```

Use R to examine the GWAS_sex_checking.sexcheck file and determine if there are individuals whose recorded sex is inconsistent with genetic sex.

```
R  
sexcheck = read.table("GWAS_sex_checking.sexcheck", header=T)  
names(sexcheck)  
sex_problem = sexcheck[which(sexcheck$STATUS=="PROBLEM"),]  
sex_problem  
q()
```

NA20530 and NA20506 were coded as a female (2) and from the genotypes appear to be males (1). In addition, 3 individuals (NA20766, NA20771 and NA20757) do not have enough information to determine if they are males or females and PLINK reports sex = 0 for the genotyped sex. Fill in the table below:

Table 1: Sex check

FID	IID	PEDSEX	SNPSEX	STATUS	F
NA20506	NA20506				
NA20530	NA20530				
NA20766	NA20766				
NA20771	NA20771				
NA20757	NA20757				

Reasons for these kinds of discrepancies, include the records are incorrect, incorrect data entry, sample swap, unreported Turner or Klinefelter syndromes. Additionally, if a sufficient number of SNPs have not been genotyped on the X chromosome it can be difficult to accurately predict the sex of an individual. In this dataset, there are only 194 X chromosomal SNPs. If you cannot validate the sex of the individual they should be removed. For this exercise, we are going to assume that when the sex

was checked, we found it was incorrectly recorded (i.e. these samples were male). Therefore, this error could simply be corrected.

Question 1: Why do you expect the homozygosity rate to be higher on the X chromosome in males than females?

c. Duplicate Samples

The following PLINK command can be used to check for duplicate samples:

```
plink --file GWAS_clean2 --genome --out duplicates
```

Open the **duplicates.genome** file in R with the following command:

```
dups = read.table("duplicates.genome", header = T)
```

We are interested in the Pi-Hat (the estimated proportion IBD sharing) value. You may notice that there is more than one duplicate (Pi-Hat \approx 1). Also, examine the output for pairs of individuals with high Pi-Hat values which can indicate they are related. The amount of allele sharing [Z(0), Z(1) and Z(2)] across all SNPs provides information on the type of relative pair.

```
problem_pairs = dups[which(dups$PI_HAT > 0.4),]  
problem_pairs
```

Table 2: Duplicate and Related Individuals

FID1	IID1	FID2	IID2	Z(0)	Z(1)	Z(2)	PI_HAT
FID1- Family ID for 1st individual; IID1 - Individual ID for 1st individual; FID2- Family ID for 2nd individual; IID2 - Individual ID for 2nd individual; Z(0)- P(IBD=0); Z(1)- P(IBD=1); Z(2)- P(IBD=2); PI_HAT-P(IBD=2)+0.5*P(IBD=1) (proportion IBD)							

Question 2: How many duplicate pairs do you find (**hint: Pi-Hat = ~1**)? Do pairs with a **Pi-Hat = ~1** have to be duplicate samples? What is another explanation? What proportion would you expect a parent/ child to share IBD? Can you find any such relationship?

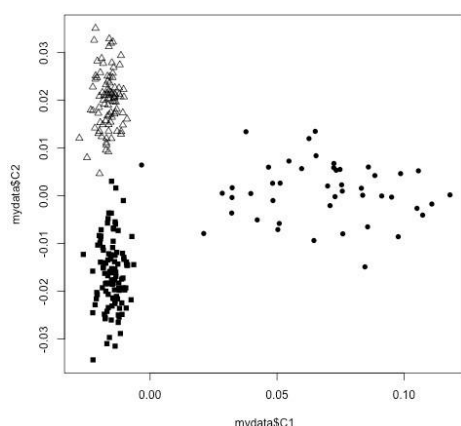
Note: Pi-hat can be inflated and individuals appear to be related to each other if you have samples from different populations. This explains why we observe pairs of individuals with Pi-hat > 0.05 since three distinct populations were analyzed. Additionally, this phenomenon can be observed if a subset(s) of samples have higher genotyping/sequencing error rates, which creates two or more “populations” and the individuals within these “populations” incorrectly appear to be related.

Using this R script please observe how many sample pairs have $\pi\text{-hat} > 0.05$:

```
problem_pairs = dups[which(dups$PI_HAT > 0.05),]  
myvars = c("FID1", "IID1", "FID2", "IID2", "PI_HAT")  
problem_pairs[myvars]
```

Create the following txt file:

1344 NA12057



name it 'IBS_excluded.txt' and save it to the folder with your PLINK data. Give the command:

```
plink --file GWAS_clean2 --remove IBS_excluded.txt
--recode --out GWAS_clean3
```

Fill in Box 4 and Oval 3.

As part of QC usually the data is examined for outliers by plotting the first and second principal or multidimensional scaling (MDS) components. Using a subset of markers that have been trimmed to remove LD ($r^2 < 0.5$). Principal components analysis (PCA) and MDS will be performed in the

second part of the exercise to detect outliers and control for populations substructure. Outlier can be due to study subjects coming from different populations e.g. European- and African-Americans or batch effects. If it is suspected that outliers are due to study subjects having been sampled from different populations than data from HapMap can be included to elucidate population membership, e.g. for a study of European-Americans if African-American study subjects are included they would cluster between the European and African HapMap samples. If you perform this type of analysis you should remove the HapMap samples and re-estimate the MDS or PC components before adjusting for population substructure or stratification. For this exercise data **is used** from HapMap Phase III which consists of CEU (Europeans from Utah), MEX (Mexicans from Los Angeles) and TSI (Tuscans from Italy). Three clusters can be observed that consist of the three data sets but no extreme outliers are observed. This data set is being used for demonstration purposes. Different populations should be analyzed separately and the results can be combined using meta-analysis. In part two of this exercise MDS and PC components will be constructed and analyzed.

d. Hardy-Weinberg Equilibrium (HWE):

To test for HWE we will test separately in each ancestry group and by case-control status. Therefore, we will need to use information on ancestry and cases-control status. Please note that this should be tested in the 3 different populations separately (CEU, MEX, TSI), but due to the small sample sizes, we tested it in the 3 populations together for example purposes. It should also be noted if the sample sizes are small it is difficult to detect a deviation from HWE.

```
plink --file GWAS_clean3 --pheno pheno.txt --pheno-name Aff --hardy
```

Using R examine the file **plink.hwe** and look for SNPs with p-values of 10^{-7} or smaller.

```
hardy = read.table("plink.hwe", header = T)
names(hardy)
hwe_prob = hardy[which(hardy$P < 0.0000009),]
hwe_prob
```

Using a criterion of $p < 10^{-7}$ to reject the null hypothesis of HWE, how many SNPs fail HWE in the controls? Fill out Oval 5 and Box 4. Using the same criteria, how many SNPs fail HWE in the controls? Complete Table 3 with this information.

Table 3: Hardy-Weinberg Equilibrium

Cases			Controls		
SNP	Pvalue	Population(s)	SNP	Population(s)	Pvalue

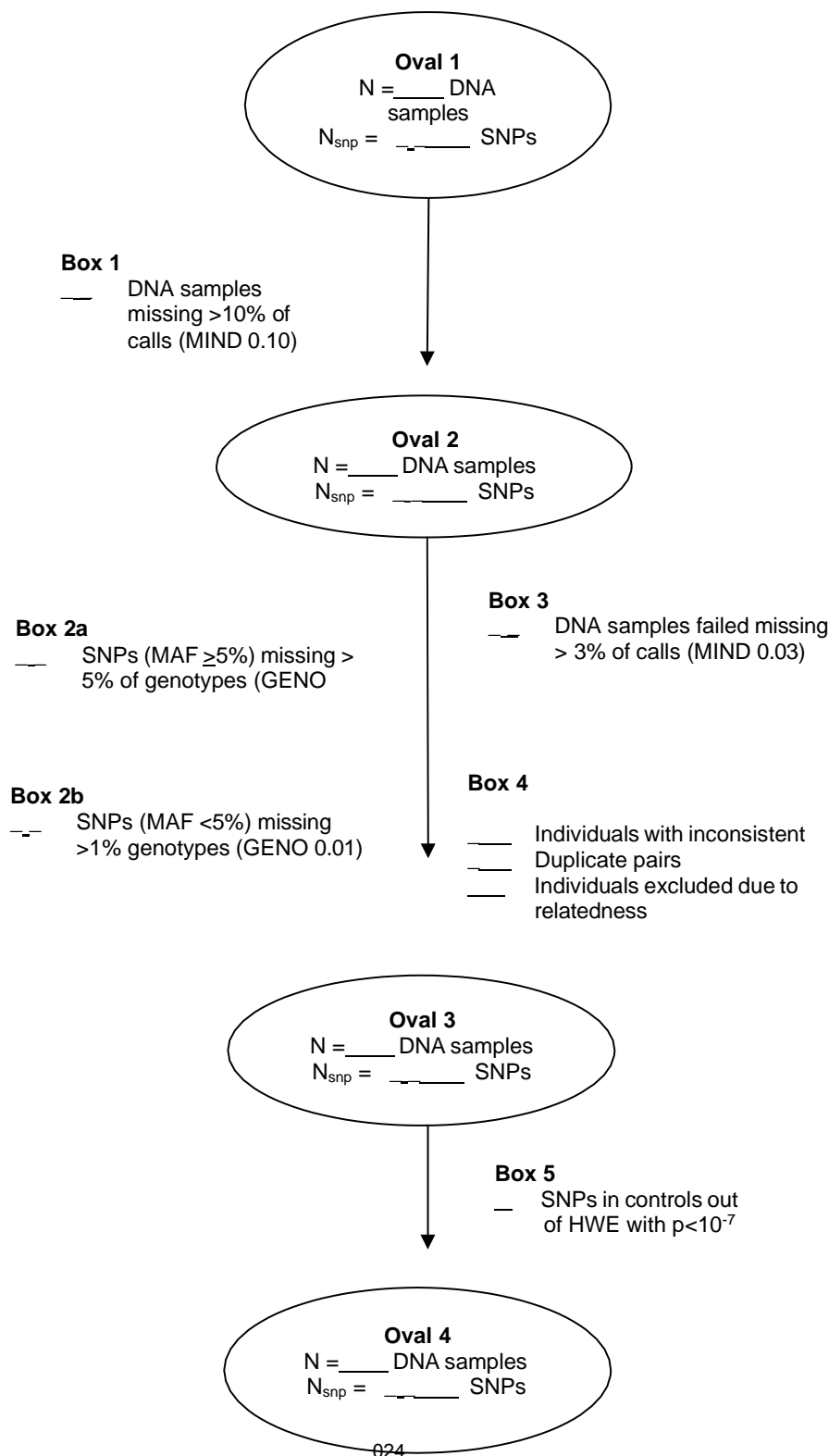
Create a text file called HWE_out.txt with the following SNP in it:

rs2968487

and type the following command:

```
plink --file GWAS_clean3 --exclude HWE_out.txt --recode --out GWAS_clean4
```

There are a number of SNPs with HWE p-values in the range of 10^{-5} to 10^{-6} in the controls. Based on above criterion they will not be excluded however, if they reach genome-wide significance during association testing they SNPs should be further investigated to ensure there is no genotyping error. You can now fill in Box 5 and Oval 4.



Answers to Questions:

Oval 1 and 2 also and Box 1 information:

```
PLINK v1.90b4.9 64-bit (13 Oct 2017)
Options in effect:
  --file GWAS
  --mind 0.10
  --out GWAS_clean_mind
  --recode

Random number seed: 1515434515
16384 MB RAM detected; reserving 8192 MB for main workspace.
Scanning .ped file... done.
Performing single-pass .bed write (6424 variants, 248 people) [Oval 1].
--file: GWAS_clean_mind-temporary.bed + GWAS_clean_mind-temporary.bim +
GWAS_clean_mind-temporary.fam written.
6424 variants loaded from .bim file.
248 people (125 males, 123 females) loaded from .fam.
1 person removed due to missing genotype data (--mind) [Box 1].
ID written to GWAS_clean_mind.irem .
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 247 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 6 het. haploid genotypes present (see GWAS_clean_mind.hh ); many
commands treat these as missing.
Total genotyping rate in remaining samples is 0.996863.
6424 variants and 247 people pass filters and QC [Oval 2].
Note: No phenotypes present.
--recode ped to GWAS_clean_mind.ped + GWAS_clean_mind.map ... done.
```

Box 2a information:

```
PLINK v1.90b4.9 64-bit (13 Oct 2017)
Options in effect:
  --file MAF_greater_5
  --geno 0.05
  --out MAF_greater_5_clean
  --recode

Random number seed: 1515435189
16384 MB RAM detected; reserving 8192 MB for main workspace.
Scanning .ped file... done.
Performing single-pass .bed write (5868 variants, 247 people).
--file: MAF_greater_5_clean-temporary.bed + MAF_greater_5_clean-temporary.bim +
MAF_greater_5_clean-temporary.fam written.
5868 variants loaded from .bim file.
247 people (125 males, 122 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 247 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 6 het. haploid genotypes present (see MAF_greater_5_clean.hh ); many
commands treat these as missing.
Total genotyping rate is 0.996858.
2 variants removed due to missing genotype data (--geno) [Box2a].
5866 variants and 247 people pass filters and QC.
Note: No phenotypes present.
--recode ped to MAF_greater_5_clean.ped + MAF_greater_5_clean.map ... done.
```

Box 2b information:

```
PLINK v1.90b4.9 64-bit (13 Oct 2017)
Options in effect:
  --file MAF_less_5
  --geno 0.01
  --out MAF_less_5_clean
  --recode

Random number seed: 1515435255
16384 MB RAM detected; reserving 8192 MB for main workspace.
```

```

Scanning .ped file... done.
Performing single-pass .bed write (556 variants, 247 people).
--file: MAF_less_5_clean-temporary.bed + MAF_less_5_clean-temporary.bim +
MAF_less_5_clean-temporary.fam written.
556 variants loaded from .bim file.
247 people (125 males, 122 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 247 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.996913.
59 variants removed due to missing genotype data (--geno) [Box2b].
497 variants and 247 people pass filters and QC.
Note: No phenotypes present.
--recode ped to MAF_less_5_clean.ped + MAF_less_5_clean.map ... done.

```

Box 3 information:

PLINK v1.90b4.9 64-bit (13 Oct 2017)

Options in effect:

```

--file GWAS_MAF_clean
--mind 0.03
--out GWAS_clean2
--recode

```

Random number seed: 1515435827

16384 MB RAM detected; reserving 8192 MB for main workspace.

Scanning .ped file... done.

Performing single-pass .bed write (6363 variants, 247 people).

```

--file: GWAS_clean2-temporary.bed + GWAS_clean2-temporary.bim +
GWAS_clean2-temporary.fam written.

```

6363 variants loaded from .bim file.

247 people (125 males, 122 females) loaded from .fam.

0 people removed due to missing genotype data (--mind) [Box 3].

Using 1 thread (no multithreaded calculations invoked).

Before main variant filters, 247 founders and 0 nonfounders present.

Calculating allele frequencies... done.

Warning: 6 het. haploid genotypes present (see GWAS_clean2.hh); many commands treat these as missing.

Total genotyping rate is 0.99716.

6363 variants and 247 people pass filters and QC.

Note: No phenotypes present.

```

--recode ped to GWAS_clean2.ped + GWAS_clean2.map ... done.

```

Answer to Question 1: Why do you expect the homozygosity rate to be higher on the X chromosome in males than females?_

Because males only have one allele for each SNP on the X chromosome they will appear homozygous.

Table 1: Sex check

FID	IID	PEDSEX	SNPSEX	STATUS	F
NA20506	NA20506	2	1	PROBLEM	1
NA20530	NA20530	2	1	PROBLEM	1
NA20766	NA20766	2	0	PROBLEM	0.2292
NA20771	NA20771	2	0	PROBLEM	0.2234
NA20757	NA20757	2	0	PROBLEM	0.2141

Table 2: Duplicate and Related Individuals

FID1	IID1	FID2	IID2	Z(0)	Z(1)	Z(2)	PI_HAT
M033	NA19774	M041	NA25000	0.0000	0.0000	1.0000	1.00
1344	NA12057	13291	NA25001	0.0000	0.0025	0.9975	1.00
1444	NA12739	1444	NA12749	0.0026	0.9807	0.0168	0.51
1444	NA12739	1444	NA12748	0.0026	0.9949	0.0025	0.50
F1D1- Family ID for 1st individual; IID1 - Individual ID for 1st individual; F1D2- Family ID for 2nd individual; IID2 - Individual ID for 2nd individual; Z(0)- P(IBD=0); Z(1)- P(IBD=1); Z(2)- P(IBD=2); PI_HAT-P(IBD=2)+0.5*P(IBD=1) (proportion IBD)							

Question 2: How many duplicate pairs do you find (**hint: Pi-Hat = ~1**)? Do pairs with a **Pi-Hat = ~1** have to be duplicate samples? What is another explanation? What proportion would you expect a parent/ child to share IBD? Can you find any such relationship?.

There are two duplicate pairs and also a trio (two parents and a child). Parent/child relationships will have a Pi_Hat value of ~0.5, but so will sibpairs. We can tell that this is a parent child relationship by examine Z(0), Z(1) and Z(2). We will retain only one sample from each duplicate pair and the parents NA12749 and NA12748. If you perform mixed-model analysis related individuals can be retained in the sample.

Oval 3 information

```
PLINK v1.90b4.9 64-bit (13 Oct 2017)
Options in effect:
  --file GWAS_clean2
  --out GWAS_clean3
  --recode
  --remove IBS_excluded.txt
Random number seed: 1515440989
16384 MB RAM detected; reserving 8192 MB for main workspace.
Scanning .ped file... done.
Performing single-pass .bed write (6363 variants, 247 people).
--file: GWAS_clean3-temporary.bed + GWAS_clean3-temporary.bim +
GWAS_clean3-temporary.fam written.
6363 variants loaded from .bim file.
247 people (125 males, 122 females) loaded from .fam.
--remove: 244 people remaining.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 244 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 6 het. haploid genotypes present (see GWAS_clean3.hh ); many commands
treat these as missing.
Total genotyping rate in remaining samples is 0.997225.
6363 variants and 244 people pass filters and QC [Oval 3].
Note: No phenotypes present.
--recode ped to GWAS_clean3.ped + GWAS_clean3.map ... done.
```

Table 3: Hardy Weinberg Equilibrium

Fail Cases		Fail Controls	
SNP	pvalue	SNP	pvalue
None		rs2968487	2.262e-007

```
PLINK v1.90b4.9 64-bit (13 Oct 2017)
Options in effect:
  --exclude HWE_out.txt
  --file GWAS_clean3
  --out GWAS_clean4
  --recode

Random number seed: 1515442367
16384 MB RAM detected; reserving 8192 MB for main workspace.
Scanning .ped file... done.
Performing single-pass .bed write (6363 variants, 244 people).
--file: GWAS_clean4-temporary.bed + GWAS_clean4-temporary.bim +
GWAS_clean4-temporary.fam written.
```

6363 variants loaded from .bim file.

244 people (123 males, 121 females) loaded from .fam.

--exclude: 6362 variants remaining.

Using 1 thread (no multithreaded calculations invoked).

Before main variant filters, 244 founders and 0 nonfounders present.

Calculating allele frequencies... done.

Warning: 6 het. haploid genotypes present (see GWAS_clean4.hh); many commands treat these as missing.

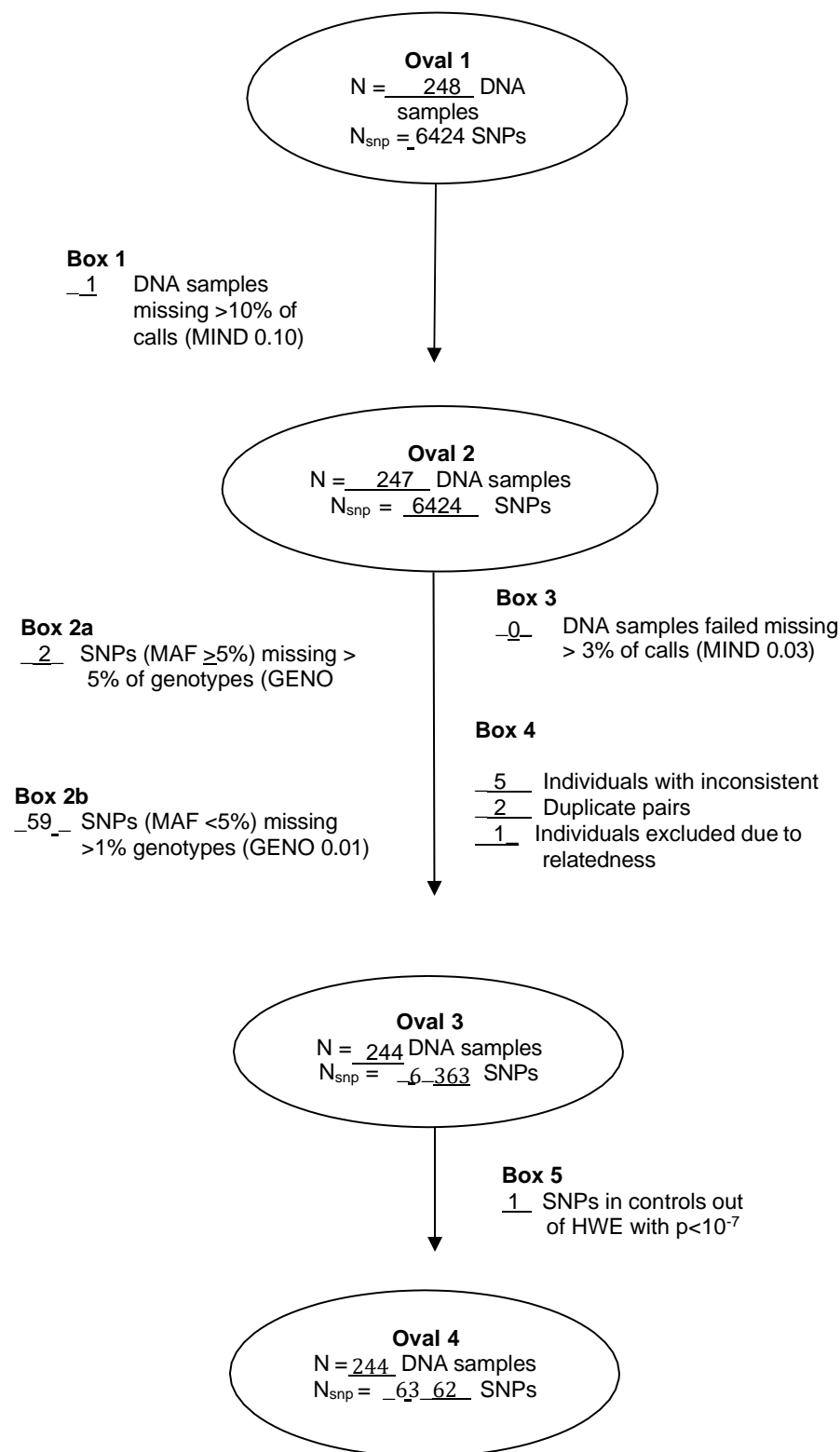
Total genotyping rate is 0.997229.

6362 variants and 244 people pass filters and QC **[Oval 4]**.

Note: No phenotypes present.

--recode ped to GW

AS_clean4.ped + GWAS_clean4.map ... done.



Exercise

Population Genetics (PP)

HWE & F statistic

From a case-control study, the following genotype counts have been observed for a SNP with alleles A and B:

Cases: $n_{\text{obs}}(\text{AA}) = 159$ $n_{\text{obs}}(\text{AB}) = 122$ $n_{\text{obs}}(\text{BB}) = 19$

Controls: $n_{\text{obs}}(\text{AA}) = 120$ $n_{\text{obs}}(\text{AB}) = 139$ $n_{\text{obs}}(\text{BB}) = 41$

Test this SNP for deviation from Hardy-Weinberg equilibrium, calculate the F_{ST} statistic in controls, test for association under a genotypic and under an allelic risk model and calculate the odds ratios for the SNP genotypes and alleles, respectively.

I. Testing for deviation from Hardy-Weinberg equilibrium (HWE) in controls

1. Calculate the genotype and allele frequencies in controls:

$$n = n_{\text{obs}}(\text{AA}) + n_{\text{obs}}(\text{AB}) + n_{\text{obs}}(\text{BB}) = \dots$$

$$f_{\text{obs}}(\text{AA}) = n_{\text{obs}}(\text{AA}) / n = \dots$$

$$f_{\text{obs}}(\text{AB}) = n_{\text{obs}}(\text{AB}) / n = \dots$$

$$f_{\text{obs}}(\text{BB}) = n_{\text{obs}}(\text{BB}) / n = \dots$$

$$f(\text{A}) = [2 \times n_{\text{obs}}(\text{AA}) + n_{\text{obs}}(\text{AB})] / 2n = \dots$$

$$f(\text{B}) = [2 \times n_{\text{obs}}(\text{BB}) + n_{\text{obs}}(\text{AB})] / 2n = \dots$$

2. Calculate the expected genotypic counts under the null hypothesis of HWE:

$$n_{\text{exp}}(\text{AA}) = n \times [f(\text{A}) \times f(\text{A})] = \dots$$

$$n_{\text{exp}}(\text{AB}) = n \times [2 \times f(\text{A}) \times f(\text{B})] = \dots$$

$$n_{\text{exp}}(\text{BB}) = n \times [f(\text{B}) \times f(\text{B})] = \dots$$

3. Arrange observed and expected genotype counts in a 2×3 table and calculate the chi-square statistic:

	AA	AB	BB
Observed (n_{obs})	120	139	41
Expected (n_{exp})

$$X^2 = [n_{\text{obs}}(\text{AA}) - n_{\text{exp}}(\text{AA})]^2 / n_{\text{exp}}(\text{AA}) + [n_{\text{obs}}(\text{AB}) - n_{\text{exp}}(\text{AB})]^2 / n_{\text{exp}}(\text{AB}) + [n_{\text{obs}}(\text{BB}) - n_{\text{exp}}(\text{BB})]^2 / n_{\text{exp}}(\text{BB})$$

$$= \dots + \dots + \dots$$

$$= \dots$$

4. Obtain the corresponding P -value from a *one*-df χ^2 distribution:

Quantiles of the 1-df χ^2 distribution using R:

```
pchisq( <<QUANTILE>>, df=1, ncp=0, lower.tail=F)
```

p =

II. Calculating the F_{ST} statistic in controls

The F_{ST} statistic can be calculated by the formula given below (introduced in the lecture on population genetics). Use the control frequencies calculated in Exercise 1 on HWE testing.

$$F_{ST} = [f_{\text{obs}}(\text{AA}) - f(\text{A}) \times f(\text{A})] / [f(\text{A}) - f(\text{A}) \times f(\text{A})]$$

$$F_{ST} = /$$

$$F_{ST} =$$

Questions

1. Is there statistical evidence at the 0.05 level that the marker is not in HWE?

2. The reported genotype counts were observed in controls only. Would it be beneficial to merge the control genotype counts with those from the cases to test HWE testing, since it would increase the sample size and power for this test? Give a reason for your answer.

3. How do you interpret this value of F_{ST} with regard to the sample (see the lecture for an interpretation of the value)?

Answers

Population Genetics (PP)

HWE & F statistic

From a case-control study, the following genotype counts have been observed for a SNP with alleles A and B:

$$\text{Cases: } n_{\text{obs}}(\text{AA}) = 159 \quad n_{\text{obs}}(\text{AB}) = 122 \quad n_{\text{obs}}(\text{BB}) = 19$$

$$\text{Controls: } n_{\text{obs}}(\text{AA}) = 120 \quad n_{\text{obs}}(\text{AB}) = 139 \quad n_{\text{obs}}(\text{BB}) = 41$$

Test this SNP for deviation from Hardy-Weinberg equilibrium, calculate the F_{ST} statistic in controls, test for association under a genotypic and under an allelic risk model and calculate the odds ratios for the SNP genotypes and alleles, respectively.

I. Testing for deviation from Hardy-Weinberg equilibrium (HWE) in controls

1. Calculate the genotype and allele frequencies in controls:

$$n = n_{\text{obs}}(\text{AA}) + n_{\text{obs}}(\text{AB}) + n_{\text{obs}}(\text{BB}) = \underline{\underline{300}}$$

$$f_{\text{obs}}(\text{AA}) = n_{\text{obs}}(\text{AA}) / n = \underline{\underline{0.400}}$$

$$f_{\text{obs}}(\text{AB}) = n_{\text{obs}}(\text{AB}) / n = \underline{\underline{0.463}}$$

$$f_{\text{obs}}(\text{BB}) = n_{\text{obs}}(\text{BB}) / n = \underline{\underline{0.137}}$$

$$f(\text{A}) = [2 \times n_{\text{obs}}(\text{AA}) + n_{\text{obs}}(\text{AB})] / 2n = \underline{\underline{0.632}}$$

$$f(\text{B}) = [2 \times n_{\text{obs}}(\text{BB}) + n_{\text{obs}}(\text{AB})] / 2n = \underline{\underline{0.368}}$$

2. Calculate the expected genotypic counts under the null hypothesis of HWE:

$$n_{\text{exp}}(\text{AA}) = n \times [f(\text{A}) \times f(\text{A})] = \underline{\underline{119.7}}$$

$$n_{\text{exp}}(\text{AB}) = n \times [2 \times f(\text{A}) \times f(\text{B})] = \underline{\underline{139.6}}$$

$$n_{\text{exp}}(\text{BB}) = n \times [f(\text{B}) \times f(\text{B})] = \underline{\underline{40.7}}$$

3. Arrange observed and expected genotype counts in a 2×3 table and calculate the chi-square statistic:

	AA	AB	BB
Observed (n_{obs})	120	139	41
Expected (n_{exp})	<u>119.7</u>	<u>139.6</u>	<u>40.7</u>

$$X^2 = [n_{\text{obs}}(\text{AA}) - n_{\text{exp}}(\text{AA})]^2 / n_{\text{exp}}(\text{AA}) + [n_{\text{obs}}(\text{AB}) - n_{\text{exp}}(\text{AB})]^2 / n_{\text{exp}}(\text{AB}) + [n_{\text{obs}}(\text{BB}) - n_{\text{exp}}(\text{BB})]^2 / n_{\text{exp}}(\text{BB})$$

$$= \underline{\underline{0.00075}} + \underline{\underline{0.00258}} + \underline{\underline{0.00221}}$$

$$= \underline{\underline{0.00554}}$$

4. Obtain the corresponding P -value from a *one*-df χ^2 distribution

Quantiles of the 1-df χ^2 distribution using R:

```
pchisq(0.00554, df=1, ncp=0, lower.tail=F)
[1] 0.9406673
```

$$p = \underline{\underline{0.94}}$$

II. Calculating the F_{ST} statistic in controls

The F_{ST} statistic can be calculated by the formula given below (introduced in the lecture on population genetics). Use the control frequencies calculated in Exercise 1 on HWE testing.

$$F_{ST} = [f_{\text{obs}}(\text{AA}) - f(\text{A}) \times f(\text{A})] / [f(\text{A}) - f(\text{A}) \times f(\text{A})]$$

$$F_{ST} = \underline{\underline{0.00010}} / \underline{\underline{0.23266}}$$

$$F_{ST} = \underline{\underline{0.00429}}$$

Questions

1. Is there statistical evidence that the marker is not in HWE?

No, the deviation from HWE is not statistically significant.

2. The reported genotype counts were observed in controls only. Would it be beneficial also to use the case genotypes for HWE testing, since it would increase the sample size for these tests? Give the reason for your answer.

No! Cases and controls have been sampled retrospectively and separately. Mixing cases and controls may cause a bias in the frequency estimates since cases are very likely oversampled, compared to their frequency in the population (unless the disease has a very high prevalence). Additionally, HWE is to be expected in the case cohort only under a multiplicative risk model.

HWE should always be tested separately in cases and controls. Inferences of potential genotyping errors by deviations from HWE should be made cautiously. For example, a deviation from HWE in cases might reflect not a genotyping error, but a genuine genetic effect. Removing that SNP because of its low HWE P -value in cases would likely reduce the power of the study!

3. How do you interpret this value of F_{ST} with regard to the sample (see lecture)?

There is a slight deficit of heterozygous genotypes for the investigated marker in the control cohort.

Linkage Disequilibrium

1.) For a 1,000 chromosomes the following haplotypes were observed.

A₁B₁ 200
A₁B₂ 50
A₂B₁ 350
A₂B₂ 400

a) What is the allele frequency for the A₁ allele and A₂ allele _____.

b.) What is the allele frequency for the B₁ and B₂ allele _____.

c.) What are the expected haplotype frequencies under linkage equilibrium

P₁₁ = A₁B₁ _____

P₁₂ = A₁B₂ _____

P₂₁ = A₂B₁ _____

P₂₂ = A₂B₂ _____

2.) Please answer the following for the above problem.

D = _____

D' = _____

r² = _____

Is there statistical evidence that Marker A and B are in linkage disequilibrium _____?

X² = _____ p-value = _____

Answers

1.) For a 1,000 chromosome the following haplotypes were observed.

A₁B₁ 200 (0.2)
A₁B₂ 50 (0.05)
A₂B₁ 350 (0.35)
A₂B₂ 400 (0.4)

a) What is the allele frequency for the A₁ allele and A₂ allele A₁=0.25 A₂=0.75.

b.) What is the allele frequency for the B₁ and B₂ allele B₁=0.55 B₂=0.45.

c.) What are the expected haplotype frequencies under linkage equilibrium?

P₁₁ = A₁B₁ 0.1375

P₁₂ = A₁B₂ 0.1125

P₂₁ = A₂B₁ 0.4125

P₂₂ = A₂B₂ 0.3375

2.) Please answer the following for the above problem.

D = (0.4*0.2)-(0.05*0.35)=0.08-0.0175=0.0625

D' = 0.0625/0.1125=0.556

r² = 0.084

Is there statistical evidence that Marker A and B are in linkage disequilibrium yes?

X² = 84.2 p-value <0.00001

Exercise

Multifactorial Analysis 1

Analyses using PLINK

In this exercise, a number of logistic regression analyses will be carried out to test for the SNP association with an affection status. This includes the adjustment for the effects of covariates and of other SNPs. Since the syntax for many of the commands is repetitive, please use the copy & paste functionality of your text editor and subsequently make the necessary changes to the copied text.

! • **Attention:** PLINK expects each command to be in a single line! PLINK ignores arguments on subsequent lines after a line break. Please type each command without a line break or use a backslash ('\') before a line break. A backslash causes PLINK to ignore the line break.

Please also answer the questions at the end of the exercise.

The data set

Please change the working directory as requested. You are provided with a data set on diastolic blood pressure and the genotypes of 20 SNP markers. The data are already in PLINK format. There are a number of files:

- **dbp.qt.*:** Set of binary PLINK files with a quantitative trait (diastolic blood pressure)
- **dbp.*:** Set of binary PLINK files with a dichotomized trait (affection status: elevated blood pressure yes/no)
- **dbp.age.pheno:** Covariate file containing the age of each individual

Use a text editor (notepad/Wordpad under Windows, pico/vi/nano/emacs under Linux) to inspect the contents of these files (except for *.bed files which are binary). Make sure you understand the meaning of each column in the files.

For this exercise, data cleaning will be skipped. First, please have a look to the questions sheet in the back. Enter the *P*-values in the table while proceeding with the exercise.

I. Logistic regression on a single SNP under an allelic model

First run a simple logistic regression analysis of all SNPs in data set:

```
plink --bfile dbp --logistic --out logreg.add
```

Inspect the result output file with a text editor:

```
logreg.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.9518	-0.4159	0.6775
...								

For each SNP, a single lines reports the relevant test (TEST), the number of non-missing observations (NMISS), the odds ratio estimate (OR), the value of the test statistic (STAT) and the corresponding *P*-value (P). Note that PLINK by default considers the *allelic* (multiplicative) model when testing for association, not the general genotypic one! The phrase ADD in the TEST column stands for an *additive* effect of the number of copies of the less-frequent allele on the *logit* scale, which is equivalent to an allelic [multiplicative] risk model. Note that other risk models could be considered using the --genotypic, --dominant, and --recessive flags. Also note that the risk model regards the A2 allele. This is not necessarily the 2 allele in the pedigree file, but simply the less frequent allele!

The `--ci` flag, used with argument `0.95`, causes PLINK to additionally calculate the 95% confidence interval for the odds ratio (OR):

```
plink --bfile dbp --logistic --ci 0.95 --out logreg.add.ci
```

Inspect the resulting file with a text editor:

```
logreg.add.ci.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	SE	L95	U95	STAT	P
11	rs1101	1021	1	ADD	600	0.9518	0.1189	0.7539	1.201	-0.4159	0.6775
...											

The file contains additional columns, containing the standard error (SE) as well as the lower (L95) and upper (U95) limits of the 95% confidence interval for the OR estimate (OR).

II. Adjustment for the effects of covariates and of other SNPs

Adjustment for the effects of covariates

Statistical analyses can be confounded by external factors. If such factors are known and measured, regression analysis allows for adjusting for their effect by simply incorporating them into the statistical model.

PLINK requires an extra file with the covariate values, for example the age of individuals. Use the text editor to inspect the covariate file `dbp.age.pheno`. Then run a logistic regression, assuming an allelic [multiplicative] model for each SNP and adjusting for the potential effect of age on the affection status:

```
plink --bfile dbp --logistic --covar dbp.age.pheno --out logreg.age.add
```

Inspect the results file with a text editor:

```
logreg.age.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.9506	-0.4262	0.67
11	rs1101	1021	1	COV1	600	1.001	0.3412	0.7329
...								

When covariates are used in the regression model, then the results file contains additional lines for each of these covariates. In this example, we have tested each marker together with a covariate named `COV1`, so we get *two* result lines per marker. The first line contains the *P*-value for the marker under an allelic model (marked by `ADD` in the `TEST` column), while the second line contains the *P*-value for the covariate (marked by `COV1` in the `TEST` column). Note that the first four columns are identical in both lines, since they relate to the same (marker-centered) regression model.

Adjusting for sex is somewhat simpler, since it is the only covariate that is contained in the pedigree file (in the 5th column). PLINK therefore only requires the `--sex` flag. Re-run the regression with an adjustment for sex and inspect the resulting file with a text editor:

```
plink --bfile dbp --logistic --sex --out logreg.sex.add
```

```
logreg.sex.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1101	1021	1	ADD	600	0.9855	-0.1201	0.9044
11	rs1101	1021	1	SEX	600	2.234	4.791	1.663e-06
...								

The covariate sex is included in the same way as a covariate above. Results are given in the line marked by `SEX` in the `TEST` column.

Finally adjust for both covariates, sex and age, simultaneously and inspect the resulting file:

```
plink --bfile dbp --logistic --sex --covar dbp.age.pheno \
      --out logreg.sexage.add
```

logreg.sexage.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.9838	-0.1345	0.893
11	rs1101	1021	1	COV1	600	1.002	0.5076	0.6118
11	rs1101	1021	1	SEX	600	2.241	4.804	1.554e-06
...								

We have now tested a marker for allele-based association while adjusting for both sex and a covariate named COV1. We thus now have *three* lines of results per marker. Potential inclusion of multiple covariates for adjustment is a great strength of the regression approach!

Adjustment for the effects of other SNPs

For many phenotypes, there are already established genetic risk factors. In many genetic epidemiological studies, one would therefore like to assess if some newly found marker association is independent of those established ones. This is equivalent to adjusting for the effect of the already established SNP. With PLINK this is achieved with the `--condition` flag.

Test all SNPs for phenotypic association under an (logit-) additive model in a logistic regression analysis while adjusting for the effect of marker rs1112:

```
plink --bfile dbp --logistic --condition rs1112 \
      --out logreg.snp1112.add
```

Inspect the results file with a text editor:

logreg.snp1112.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.9607	-0.3278	0.7431
11	rs1101	1021	1	rs1112	600	2.149	5.636	1.738e-08
...								

Marker rs1112 is considered a covariate in the association analysis and the **TEST** column is correspondingly named. The **ADD** line contains the *P*-value for the tested SNP marker, while **rs1112** contains the *P*-value for that SNP for which the analysis is adjusted. Note that the marker for whom the regression model is adjusted (here: rs1112) is *always* considered to act under an additive (really: multiplicative) risk model!

Now run the same analysis, but this time adjusting for marker rs1117. Inspect the results file with a text editor:

```
plink --bfile dbp --logistic --condition rs1117 \
      --out logreg.snp1117.add
```

logreg.snp1117.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.9468	-0.4501	0.6527
11	rs1101	1021	1	rs1117	600	2.226	4.865	1.142e-06
...								

III. Analysis of quantitative instead of dichotomized trait

Trait values are often dichotomized. For example, blood pressure values above a certain threshold could be declared as ‘elevated’ while those below would be considered ‘normal’. Dichotomization can result in a power loss, because information is discarded. In our example data set, the original trait value (diastolic blood pressure) had been dichotomized to case-control status using some threshold.

The file `dbp.qt.ped` contains the original *quantitative* trait values, which are approximately normally distributed. Run a *linear regression* analysis with PLINK with adjustment for the effect of sex:

```
plink --bfile dbp.qt --linear --sex --out linreg.sex.add
```

Inspect the results file: linreg.sex.add.assoc.linear

CHR	SNP	BP	A1	TEST	NMISS	BETA	STAT	P
11	rs1101	1021	1	ADD	600	-0.05063	-0.1401	0.8887
11	rs1101	1021	1	SEX	600	3.02	6.05	2.553e-09
...								

Since we perform a linear regression analysis, PLINK now reports the regression coefficient (slope) b , not an odds ratio. Also, since we adjust for the covariate sex in this example, we get two result lines per marker. The allelic model in a linear regression analysis is equivalent to an additive model.

Questions

1. Please enter the P -values for marker rs1112 from the analyses in the table below.

	Type of analysis	P -value
I.	Single marker, case-control, allelic model	
II.	Single marker, case-control, adjustment for age	
	Single marker, case-control, adjustment for sex	
	Single marker, case-control, adjustment for sex & age	
	Single marker, case-control, adjustment for marker rs1117	
III.	Single marker, quantitative trait, adjustment for sex	

2. Please give the odds ratio (OR) and its 95% confidence interval for marker rs1112 in the unadjusted case-control analysis.

OR =
95% CI = -

3. The P -value for the quantitative-trait analysis is much smaller than for the case-control analysis. Do you have an explanation?

Analyses using R

In this exercise, a number of logistic and linear regression analyses will be carried out to test for the association of a number of SNPs with an affection status and with a quantitative trait, respectively. This includes the use of different tests, the calculation of odds ratios (OR), and the consideration of different genetic models. Further objectives are the adjustment for the effects of covariates and the testing of a SNP for association given the effect of another SNP.

The data set is the same as with the PLINK exercise. For convenience, it has already been converted to R format and stored in the file `dbp.R`.

Since the syntax for many of the commands is highly repetitive and in order to save time, please use the copy & paste functionality of your text editor and subsequently make the necessary changes to the copied text.

Please also answer the questions at the end of the exercise.

Data set import

Start R and change the working directory as requested. Load the data set for the exercise and get an overview which objects have been loaded into the R working memory:

```
load("dbp.R")
ls()
dbp[1:5,]
summary(dbp)
```

I. Logistic regression on a single SNP genotype

Logistic regression models in R are implemented through the `glm` function. This function requires a model formulation. This includes a specification of what is regressed on what (e.g. `affection ~ rs1112`), the error family, the link function, and the data set to be used.

Run a logistic regression analysis of the affection status regressed on the genotype of marker `rs1112`, using the data in the data frame `dbp`. Assign the results from the regression analysis to the new object `result.snp12`:

```
result.snp12 = glm (affection ~ rs1112, family=binomial("logit"), data=dbp)
```

Print the results of the regression analysis with the following command:

```
print (result.snp12)
```

The marker variable `rs1112` is of data type `factor` (nominal). Thus, we have considered a general *genotypic* model. R has therefore created two dummy variables, named `rs11123` and `rs11124`, which separately describe the effects of the genotypes coded as **3** (heterozygous 1/2) and **4** (homozygous 2/2), respectively. The effects of these two genotypes are compared to the baseline genotype **2** (homozygous 1/1).

The results object is part of some special R classes, namely `lm` and `glm` (same names as the functions). Membership in these classes causes R to use dedicated, specialized functions for printing, analyzing and other tasks with such objects:

```
print ( class (result.snp12) )
print ( summary(result.snp12) )
```

The coefficients table lists the estimated values for the regression coefficients β as well as their standard errors. It further contains the *P*-values as obtained from a Wald test.

To carry out a likelihood-ratio test (LRT), first calculate the χ^2 statistic and subsequently obtain the corresponding *P*-value. Note that we have a χ^2 distribution with *two* degrees of freedom, since we test two dummy variables simultaneously against the null model:

```
dev.geno = anova (result.snp12, test="Chi")
lrt.pvalue = pchisq(dev.geno[dim(dev.geno)[1], "Deviance"],
                    df=2, ncp=0, FALSE)
print ( lrt.pvalue )
```

We can also access parts of the results object with indexes. For example, we can extract the regression coefficients and calculate the odds ratios for the genotypes (reminder from the lecture: $OR=e^{\beta}$) as well as their confidence intervals:

```
print ( summary(result.snp12)$coefficients )
snp.beta = summary(result.snp12)$coefficients[2:3,1]
print ( snp.beta )
print ( exp(snp.beta) )

ci = confint (result.snp12)
print (ci)
print ( exp(ci) )
```

So far, the marker data are of type `factor` (nominal) and we have considered a general genotypic model. For an allelic (multiplicative) model, the data type has to be changed to `numeric`. This way, the genotype is recoded from nominal 2/3/4 (for 11/12/22) to numeric 0/1/2 (for the number of copies of the “2” allele with each sample):

```
snp.data = dbp[,c("affection", "rs1112")]
```

```
summary(snp.data)

snp.data[, "rs1112"] <- as.numeric(snp.data[, "rs1112"]) - 1
summary(snp.data)
```

Run the logistic regression analysis again, this time assuming an allelic model:

```
result.all = glm (affection ~ rs1112, family=binomial("logit"),
                  data=snp.data)
dev.all     = anova (result.all, test="Chi")
summary(result.all)
print(dev.all)
```

II. Adjustment for the effects of covariates and of other SNPs

Analyses can be confounded by external factors. If such factors are known and measured, regression analysis allows for adjusting for their effect by simply incorporating them into the statistical model.

First, create an excerpt from the full data set. For all subsequent analyses, we will consider an allelic (multiplicative) model for the markers:

```
snp.data = dbp[,c("affection", "trait", "sex", "age", "rs1112", "rs1117")]
summary(snp.data)

snp.data[, "rs1112"] <- as.numeric(snp.data[, "rs1112"]) - 1
snp.data[, "rs1117"] <- as.numeric(snp.data[, "rs1117"]) - 1
```

Adjustment for the effects of covariates

Does sex have an effect on the affection status and is the effect of the SNP independent of such a potential influence? To answer this question, re-run the regression analysis for SNP rs1112, this time with an adjustment for sex:

```
result.adj = glm (affection ~ sex + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

Age is also often suspected to influence the trait of interest. Therefore, re-run the analysis with an adjusting for sample age:

```
result.adj = glm (affection ~ age + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

Finally, adjust for both covariates, sex and age, simultaneously in the regression analysis:

```
result.adj = glm (affection ~ sex + age + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

Adjustment for the effects of other SNPs

For many diseases and phenotypes, there are already established genetic factors. In many genetic epidemiological studies, one would therefore like to assess if some newly found association is independent of such established ones. This is equivalent to adjusting for the effect of the already established SNP.

Run a logistic regression analysis for each of the two SNPs rs1112 and rs1117, while adjusting for the effect of the other:

```
result.adj = glm (affection ~ rs1117 + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
anova (result.adj, test="Chi")
```

```
result.adj = glm (affection ~ rs1112 + rs1117, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
anova (result.adj, test="Chi")
```

Note that the *P*-values from a Wald test do not differ for the different orders of markers, but that the *P*-values from a likelihood-ratio test (obtained from the `anova` function) do!

III. Analysis of quantitative instead of dichotomized trait

Dichotomization of quantitative trait values can result in a power loss, because information is discarded. In our example data set, the original trait value (diastolic blood pressure) had been dichotomized to case-control status: All individuals with a value greater than a certain threshold were defined as having high blood pressure (“cases”), whereas the others were considered to be controls with normal blood pressure.

The column `trait` in the data frame `dbp` contains the original quantitative trait values. Run two linear regression analyses, one without and one with adjust for the effect of sex:

```
result.adj = lm (trait ~ rs1112, data=snp.data)
summary(result.adj)

result.adj = lm (trait ~ sex + rs1112, data=snp.data)
summary(result.adj)
```

Quitting

Quit the R session by calling the `quit` function:

```
q()
```

Questions

1. Please enter the *P*-values for marker `rs1112` from the analyses in the table below.

	Type of Analysis	<i>P</i> -value
I.	Single marker, case-control, genotypic model	Wald test: het 1/2: hom 2/2: LRT:
	Single marker, case-control, allelic model	Wald: LRT:
II.	Single marker, case-control, adjustment for age	
	Single marker, case-control, adjustment for sex	
	Single marker, case-control, adjustment for sex & age	
	Single marker, case-control, adjustment for marker <code>rs1117</code>	Wald: LRT:
III.	Single marker, quantitative trait, adjustment for sex	

2. Please give the odds ratio (OR) and its 95% confidence interval for marker rs1112 in the unadjusted, genotype-based case-control analysis.

OR_{het(1/2)} = 95% CI = -
 OR_{hom(2/2)} = 95% CI = -

3. The *P*-values obtained from R slightly differ from those obtained from PLINK. Do you have an explanation?

4. In the combined analysis of rs1112 and rs1117 (section II., no interaction), the LRT-based *P*-values strongly depended on the order of the markers in the regression model (affection ~ rs1117+rs1112: $p_{rs1117}=5.547e-07$ / $p_{rs1112}=1.193e-03$; affection ~ rs1112+rs1117: $p_{rs1112}=5.438e-09$ / $p_{rs1117}=0.21$). Do you have an explanation?

Answers

Multifactorial Analysis 1

Analyses using PLINK

I. Logistic regression on a single SNP under an allelic model

```
plink --bfile dbp --logistic --out logreg.add
```

```
logreg.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1111	1245388	2	ADD	600	1.041	0.3465	0.729
11	rs1112	1245604	2	ADD	600	2.149	5.642	1.683e-08
11	rs1113	1246723	2	ADD	600	1.654	3.744	0.0001809
...								
11	rs1117	1258119	2	ADD	600	2.224	4.864	1.151e-06
...								

Each line contains the result for the marker whose name is given in column 'SNP'. The 'TEST' column contains an 'ADD' for an 'additive', or more precisely, 'allelic' model. Under a logistic scale, the allelic model is equivalent to a multiplicative model. Column 'NMISS' contains the number of observations used for the test, while 'OR' and 'P' contain the odds ratio and the nominal *P*-value.

Note that the A1 column contains the minor, or risk, allele. The considered genetic risk model and the reported odds ratio are with regard to this allele! For example, marker rs1112 has been tested under an allele-based model ('ADD') for minor allele '2' ($p=1.7 \times 10^{-8}$). In the context of the used logistic regression analysis, this corresponds to a multiplicative risk model for this allele. Heterozygous carriers of the '2' risk allele have an increased risk of $\psi_1=2.15$ (approximated by the odds ratio) compared to the baseline genotype (homozygous for the major allele), while homozygous carriers of '2' have an increased risk of $\psi_2=4.62$ (multiplicative risk model!).

logreg.add.ci.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	SE	L95	U95	STAT	P
...											
11	rs1111	1245388	2	ADD	600	1.041	0.1155	0.8299	1.305	0.3465	0.729
11	rs1112	1245604	2	ADD	600	2.149	0.1356	1.648	2.804	5.642	1.683e-08
11	rs1113	1246723	2	ADD	600	1.654	0.1344	1.271	2.153	3.744	0.0001809
...											
11	rs1117	1258119	2	ADD	600	2.224	0.1643	1.612	3.069	4.864	1.151e-06
...											

The two additional columns ‘L95’ and ‘U95’ contain the lower and upper limit of the 95% confidence interval.

II. Adjustment for the effects of covariates and of other SNPs

Adjustment for the effects of covariates

```
plink --bfile dbp --logistic --covar dbp.age.pheno --out logreg.age.add
```

logreg.age.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1112	1245604	2	ADD	600	2.149	5.642	1.681e-08
11	rs1112	1245604	2	COV1	600	1.001	0.3289	0.7423
...								

When covariates are used in the regression model, then the results file contains additional lines for each of these covariates. In this example, we have tested each marker together with a covariate named COV1, so we get *two* result lines per marker. The first line contains the *P*-value for the marker under an allelic model ($p=1.68 \times 10^{-8}$), while the second line contains the *P*-value for the covariate ($p=0.74$).

```
plink --bfile dbp --logistic --sex --out logreg.sex.add
```

logreg.sex.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1112	1245604	2	ADD	600	2.163	5.574	2.495e-08
11	rs1112	1245604	2	SEX	600	2.257	4.719	2.373e-06
...								

The covariate sex is included in the same way as a covariate above. Results are given in the line marked by SEX in the TEST column.

```
plink --bfile dbp --logistic --sex --covar dbp.age.pheno \
--out logreg.sexage.add
```

logreg.sexage.add.assoc.logistic

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1112	1245604	2	ADD	600	2.163	5.574	2.485e-08
11	rs1112	1245604	2	COV1	600	1.002	0.5076	0.6117
11	rs1112	1245604	2	SEX	600	2.265	4.733	2.21e-06
...								

We have now tested a marker for allele-based association while adjusting for both sex and a covariate named COV1. We thus now have *three* lines of results per marker. For example for marker rs1112, sex shows a significant association with the phenotype ($p=2.21 \times 10^{-6}$), but not the covariate ($p=0.61$). After adjustment for these covariates, this marker still shows a significant association ($p=2.48 \times 10^{-8}$).

Adjustment for the effects of other SNPs

```
plink --bfile dbp --logistic --condition rs1112 \
```

```
--out logreg.snp1112.add
logreg.snp1112.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1117	1258119	2	ADD	600	1.33	1.242	0.2143
11	rs1117	1258119	2	rs1112	600	1.822	3.186	0.00144
...								

Here, we have tested for a phenotypic association of marker rs1117 and adjusted for the covariate marker rs1112. After this adjustment, the marker rs1117 does not show a significant association ($p=0.21$).

```
plink --bfile dbp --logistic --condition rs1117 \
      --out logreg.snp1117.add
logreg.snp1117.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1112	1245604	2	ADD	600	1.822	3.186	0.00144
11	rs1112	1245604	2	rs1117	600	1.33	1.242	0.2143
...								

Here, we have tested for a phenotypic association of marker rs1112 and adjusted for the covariate marker rs1117. Marker rs1112 is significantly associated ($p=0.0014$) after adjustment for effects of marker rs1117.

III. Analysis of quantitative instead of dichotomized trait

```
plink --bfile dbp.qt --linear --sex --out linreg.sex.add
linreg.sex.add.assoc.linear
```

CHR	SNP	BP	A1	TEST	NMISS	BETA	STAT	P
...								
11	rs1112	1245604	2	ADD	600	2.868	7.82	2.406e-14
11	rs1112	1245604	2	SEX	600	2.882	6.071	2.268e-09
...								

Since we adjust for the covariate sex, we get two result lines per marker. Marker rs1112 shows a highly significant association with the phenotype. Since we perform a linear analysis, PLINK now reports the regression coefficient β , not an odds ratio. The allelic model in a linear regression analysis is equivalent to an additive model.

Questions

1. Please enter the P -values for marker rs1112 from the analyses in the table below.

	Type of analysis	P -value
I.	Single marker, case-control, allelic model	1.683e-08
II.	Single marker, case-control, adjustment for age	1.681e-08
	Single marker, case-control, adjustment for sex	2.495e-08
	Single marker, case-control, adjustment for sex & age	2.485e-08
	Single marker, case-control, adjustment for marker rs1117	0.00144
III.	Single marker, quantitative trait, adjustment for sex	2.406e-14

2. Please give the odds ratio (OR) and its 95% confidence interval for marker rs1112 in the unadjusted case-control analysis.

OR = 2.149
95% CI = 1.648 – 2.804

3. The *P*-value for the quantitative-trait analysis is much smaller than for the case-control analysis. Do you have an explanation?

Dichotomizing quantitative values can lead to a considerable information loss. This has been the case here. The new binary trait contains the information that a value is below or above a certain threshold, but it does not quantify *how much greater or smaller* than this threshold this value is, i.e. the information on the distance of the value from this threshold is lost. If such information can be used in the statistical analysis, methods can be more powerful and can lead to smaller *P*-values.

Analyses using R

I. Logistic regression on a single SNP genotype

```
# --- Regression + Wald --- #
result.snp12 = glm (affection ~ rs1112, family=binomial("logit"), data=dbp)
print (result.snp12)
Call:  glm(formula = affection ~ rs1112, family = binomial("logit"),      data = dbp)
Coefficients:
(Intercept)      rs11123      rs11124
      -0.4449       0.7582       1.5435

Degrees of Freedom: 599 Total (i.e. Null);  597 Residual
Null Deviance:      831.8
Residual Deviance: 797.7      AIC: 803.7

print ( class (result.snp12) )
[1] "glm" "lm"

print ( summary(result.snp12) )
Call:
glm(formula = affection ~ rs1112, family = binomial("logit"),
    data = dbp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6651  -0.9952  -0.1183   1.0476   1.3712

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4449     0.1189  -3.741 0.000183 ***
rs11123       0.7582     0.1746   4.343 1.40e-05 ***
rs11124       1.5435     0.3416   4.518 6.24e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 797.75  on 597  degrees of freedom
AIC: 803.75

Number of Fisher Scoring iterations: 4
```

The SNP genotype of marker rs1112 has been stored as `factor` type in R. Since there is no distance defined between categories, R has automatically defined two dummy variables, namely rs11123 and rs11124, to code for the presence or absence of the '3' and of the '4' genotypes, respectively. The dummy variable assumes the value 1 if the respective genotype is present in the particular individual and 0 otherwise. If both '3' and '4' are absent, then the baseline genotype '2' is assumed to be present. Using a Wald test, each of the dummy variables has been tested separately for significant association with the phenotype (affection status). Both variables are significantly associated with the phenotype.

```
# --- Likelihood-ratio test --- #
dev.gen0 = anova (result.snp12, test="Chi")
lrt.pvalue = pchisq(dev.gen0[dim(dev.gen0)[1], "Deviance"],
                    df=2, ncp=0, FALSE)
print ( lrt.pvalue )
[1] 4.077856e-08
```

Often, we are not interested in the effect of a particular genotype, e.g. '3' or '4', but in the overall significance of a marker. To test this, we have to compare the null model (without both dummy variables) against the alternative model (with both dummy variables) using a likelihood-ratio test. Because the two models differ in two parameters, we have to compare the deviance against a χ^2 distribution with two parameters.

```
# --- OR + CI --- #
print ( summary(result.snp12)$coefficients )
      Estimate Std. Error   z value    Pr(>|z|)
(Intercept) -0.4449068  0.1189351 -3.740754 1.834691e-04
rs11123      0.7582015  0.1745740  4.343154 1.404519e-05
rs11124      1.5435191  0.3416277  4.518132 6.238747e-06

# Coefficients (betas) for the both dummy variables #
snp.beta = summary(result.snp12)$coefficients[2:3,1]
print ( snp.beta )
      rs11123  rs11124
0.7582015 1.5435191
# Odds ratios (OR) for both dummy variables [OR=exp(beta)] #
print ( exp(snp.beta) )
rs11123  rs11124
2.134434 4.681034
```

```
# 95% confidence interval for betas #
ci = confint (result.snp12)
Waiting for profiling to be done...
print (ci)
```

```
      2.5 %      97.5 %
(Intercept) -0.6802726 -0.2135169
rs11123      0.4176220  1.1023701
rs11124      0.8984800  2.2475097
```

```
# 95% confidence intervals for OR #
print ( exp(ci) )
```

```
      2.5 %      97.5 %
(Intercept) 0.5064789 0.8077385
rs11123      1.5183466 3.0112947
rs11124      2.4558674 9.4641382
```

```
# --- Allelic model --- #
snp.data = dbp[,c("affection", "rs1112")]
summary(snp.data)
affection rs1112
0:300      2:297
1:300      3:251
         4: 52
```

```
snp.data[,"rs1112"] <- as.numeric(snp.data[,"rs1112"]) - 1
summary(snp.data)
affection      rs1112
0:300      Min.    :0.0000
1:300      1st Qu.:0.0000
           Median :1.0000
           Mean   :0.5917
           3rd Qu.:1.0000
           Max.    :2.0000
```

Because we have coded the marker genotype as numeric, another summary than for factor data is provided.

```
# --- Allelic model for allele 2 of marker rs1112 --- #
result.all = glm (affection ~ rs1112, family=binomial("logit"),
                  data=snp.data)
dev.all     = anova (result.all, test="Chi")
summary(result.all)
Call:
glm(formula = affection ~ rs1112, family = binomial("logit"),
    data = snp.data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6582  -0.9944  -0.1154   1.0456   1.3722
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4470     0.1142  -3.913 9.10e-05 ***
rs1112         0.7652     0.1356   5.642 1.68e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 797.75 on 598 degrees of freedom
AIC: 801.75
```

Number of Fisher Scoring iterations: 4

```
print(dev.all)
Analysis of Deviance Table

Model: binomial, link: logit
Response: affection
Terms added sequentially (first to last)
```

	Df	Deviance	Resid.	Df	Resid.	Dev	P(> Chi)
NULL			599		831.78		
rs1112	1	34.03	598		797.75	5.438e-09	

Due to the numeric coding of the genotype, there is now an interpretable distance between 0, 1 and 2 copies of the second allele. This allele count can readily enter the regression model, without any need for creating dummy variables. With large-enough sample sizes, the P-values from the Wald test and from the likelihood-ratio test are very similar.

II. Adjustment for the effects of covariates and of other SNPs

```
# --- Data conversion for all subsequent analyses --- #
snp.data = dbp[,c("affection", "trait", "sex", "age", "rs1112", "rs1117")]
summary(snp.data)
affection      trait      sex      age      rs1112  rs1117
0:300      Min.    : 60.50    1:329    Min.    :18.00    2:297    2:396
1:300      1st Qu.: 77.44    2:271    1st Qu.:38.00    3:251    3:190
           Median : 82.00           Median :55.00    4: 52    4: 14
           Mean   : 81.85           Mean   :55.49
           3rd Qu.: 86.09           3rd Qu.:74.00
           Max.   :101.49           Max.   :90.00

snp.data[,"rs1112"] <- as.numeric(snp.data[,"rs1112"]) - 1
snp.data[,"rs1117"] <- as.numeric(snp.data[,"rs1117"]) - 1
```

For the subsequent analysis, we convert the marker genotypes to numeric coding, i.e. we will consider an allele-based model.

Adjustment for the effects of covariates

```
# Adjustment for sex #
result.adj = glm (affection ~ sex + rs1112      , family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
Call:
glm(formula = affection ~ sex + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.82645  -1.12415  -0.09007   1.21323   1.57462

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.08386    0.13730  -0.611    0.541
sex2         -0.81412    0.17253  -4.719 2.37e-06 ***
rs1112        0.77139    0.13840   5.574 2.49e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 774.98 on 597 degrees of freedom
AIC: 780.98

Number of Fisher Scoring iterations: 4
```

Compared to males, females have a significantly decreased risk ($p=2.4 \times 10^{-6}$; $OR=e^{-0.814}=0.44$) of becoming affected. The sex-adjusted P -value for marker rs1112 is highly significant ($p=2.5 \times 10^{-8}$), which causes an increase in risk of $e^{0.771}=2.16$ for heterozygous carriers and of $2.16^2=4.67$ for homozygous carriers of the risk allele (allele-based, i.e. multiplicative risk model!).

```
# Adjustment for age #
result.adj = glm (affection ~ age + rs1112      , family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ age + rs1112, family = binomial("logit"),
     data = snp.data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6776  -1.0066  -0.1132   1.0550   1.3937
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.520422   0.250956  -2.074   0.0381 *
age          0.001322   0.004020   0.329   0.7423
rs1112       0.765189   0.135624   5.642 1.68e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 797.64 on 597 degrees of freedom
AIC: 803.64
```

Number of Fisher Scoring iterations: 4

Adjustment for sex and age

```
result.adj = glm (affection ~ sex + age + rs1112, family=binomial("logit"),
                  data=snp.data)
```

```
summary(result.adj)
```

```
Call:
glm(formula = affection ~ sex + age + rs1112, family = binomial("logit"),
     data = snp.data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.84985  -1.12493  -0.08714   1.19367   1.60989
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.198133   0.263732  -0.751   0.452
sex2        -0.817603   0.172736  -4.733 2.21e-06 ***
age          0.002084   0.004105   0.508   0.612
rs1112       0.771546   0.138411   5.574 2.48e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 774.72 on 596 degrees of freedom
AIC: 782.72
```

Number of Fisher Scoring iterations: 4

Age has no significant impact on the phenotype, while sex does. Marker rs1112 is highly significantly associated with affection status ($p=2.5 \times 10^{-8}$) after adjusting for the effects of the covariates sex and age.

Adjustment for the effects of other SNPs

```
# Association analysis of rs1112, adjusted for the effects of rs1117 #
result.adj = glm (affection ~ rs1117 + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ rs1117 + rs1112, family = binomial("logit"),
    data = snp.data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7636  -0.9923  -0.1518   1.1154   1.3745
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4523     0.1144  -3.955 7.66e-05 ***
rs1117         0.2853     0.2297   1.242  0.21431
rs1112         0.5999     0.1883   3.186  0.00144 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 796.21 on 597 degrees of freedom
AIC: 802.21
```

Number of Fisher Scoring iterations: 4

```
anova (result.adj, test="Chi")
Analysis of Deviance Table
```

```
Model: binomial, link: logit
Response: affection
Terms added sequentially (first to last)
```

	Df	Deviance	Resid. Df	Resid. Dev	P(> Chi)
NULL			599	831.78	
rs1117	1	25.06	598	806.71	5.547e-07
rs1112	1	10.50	597	796.21	1.193e-03

The Wald test compares the model with the predictor (here: rs1117) against the model without it; marker rs1112 is included in both models. The ANOVA sequence procedure first applied a likelihood-ratio test for the null model (no predictors) against the model that only includes rs1117 ($p=5.5\times 10^{-7}$) and subsequently compares the latter model against the one that includes rs1117 and rs1112 ($p=1.2\times 10^{-3}$). Thus, it first assesses the contribution of rs1117 and subsequently the *additional* contribution of rs1112.

```
# Association analysis of rs1117, adjusted for the effects of rs1112 #
result.adj = glm (affection ~ rs1112 + rs1117, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
Call:
glm(formula = affection ~ rs1112 + rs1117, family = binomial("logit"),
    data = snp.data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7636  -0.9923  -0.1518   1.1154   1.3745
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4523     0.1144  -3.955 7.66e-05 ***
rs1112         0.5999     0.1883   3.186  0.00144 **
rs1117         0.2853     0.2297   1.242  0.21431
```

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

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 796.21 on 597 degrees of freedom
AIC: 802.21
```

Number of Fisher Scoring iterations: 4

anova (result.adj, test="Chi")

Analysis of Deviance Table

Model: binomial, link: logit

Response: affection

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	P(> Chi)
NULL			599	831.78	
rs1112	1	34.03	598	797.75	5.438e-09
rs1117	1	1.54	597	796.21	0.21

The Wald test again compares the model with the predictor (here: rs1117) against the model without it; marker rs1112 is included in both models. Results are therefore the identical for the models `affection ~ rs1117 + rs1112` (see above) and `affection ~ rs1112 + rs1117`. However, the ANOVA sequence procedure (using a likelihood-ratio test) now first compares model `affection ~ constant` (null model) against `affection ~ rs1112` and only includes rs1117: `affection ~ rs1112 + rs1117`. Marker rs1112 makes a significant contribution ($p=5.4 \times 10^{-9}$). On top of this, marker rs1117 does not contain any new information and does not provide a significant additional contribution ($p=0.2$).

The explanation for this observation is that both markers, rs1112 and rs1117, are in linkage disequilibrium with each other and also with the causal genetic variant, but that marker rs1112 shows the higher allelic correlation with that variant. If marker rs1117 is first included in the regression model, then marker rs1112 can still provide some additional association information. However, if marker rs1112 is first included, then marker rs1117 has nothing to offer and will be insignificant.

III. Analysis of quantitative instead of dichotomized trait

```
# --- Single-marker analysis with *linear* model --- #
```

```
result.adj = lm (trait ~ rs1112 , data=snp.data)
```

```
summary(result.adj)
```

```
Call:
```

```
lm(formula = trait ~ rs1112, data = snp.data)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-22.5556  -3.9106   0.2194   4.0144  15.4809
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	80.1021	0.3301	242.680	< 2e-16 ***
rs1112	2.9535	0.3774	7.826	2.29e-14 ***

```
---
```

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

```
Residual standard error: 5.954 on 598 degrees of freedom
```

```
Multiple R-squared:  0.09291, Adjusted R-squared:  0.09139
```

```
F-statistic: 61.25 on 1 and 598 DF, p-value: 2.292e-14
```

The genotype of marker rs1112 has been coded as `numeric`; an allele-based test has therefore been performed. In a linear regression model, this corresponds to an *additive* risk model. Marker rs1112 is highly associated with the quantitative trait (Wald test: $p=2.3 \times 10^{-14}$). The linear regression model is able to explain about 9% of the observed variance in the quantitative trait ($R^2=0.093$).

```
# --- Additional adjustment for sex --- #
result.adj = lm (trait ~ sex + rs1112, data=snp.data)
summary(result.adj)
Call:
lm(formula = trait ~ sex + rs1112, data = snp.data)

Residuals:
    Min       1Q   Median       3Q      Max
-20.9404  -3.6272   0.2234   3.7815  16.3480

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  81.4542     0.3904  208.654 < 2e-16 ***
sex2         -2.8823     0.4748  -6.071 2.27e-09 ***
rs1112        2.8685     0.3668   7.820 2.41e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 5.784 on 597 degrees of freedom
Multiple R-squared:  0.1456, Adjusted R-squared:  0.1428
F-statistic: 50.89 on 2 and 597 DF, p-value: < 2.2e-16
```

Marker rs1112 is highly associated with the quantitative trait after adjusting for the effect of sex (Wald test: $p=2.4 \times 10^{-14}$; likelihood-ratio test: $p < 2.2 \times 10^{-16}$). The linear regression model is now able to explain more than 14% of the observed trait variance, i.e. inclusion of sex provides a better model fit to the sample data.

Questions

1. Please enter the *P*-values for marker rs1112 from the analyses in the table below.

	Type of Analysis	<i>P</i> -value
I.	Single marker, case-control, genotypic model	Wald test: het 1/2: 1.40e-05 hom 2/2: 6.24e-06 LRT: 4.077856e-08
	Single marker, case-control, allelic model	Wald: 1.68e-08 LRT: 5.438e-09
II.	Single marker, case-control, adjustment for age	1.68e-08
	Single marker, case-control, adjustment for sex	2.49e-08
	Single marker, case-control, adjustment for sex & age	2.48e-08
	Single marker, case-control, adjustment for marker rs1117	Wald: 0.00144 LRT: 1.193e-03
III.	Single marker, quantitative trait, adjustment for sex	2.29e-14

2. Please give the odds ratio (OR) and its 95% confidence interval for marker rs1112 in the unadjusted, genotype-based case-control analysis.

$$\begin{aligned}\text{OR}_{\text{het}(1/2)} &= 2.134434 & 95\% \text{ CI} &= 1.5183466 - 3.0112947 \\ \text{OR}_{\text{hom}(2/2)} &= 4.681034 & 95\% \text{ CI} &= 2.4558674 - 9.4641382\end{aligned}$$

3. The *P*-values obtained from R slightly differ from those obtained from PLINK. Do you have an explanation?

PLINK and R can slightly differ in the *numerical* implementation of the same statistical tests. However, the resulting differences are minor.

4. In the combined analysis of rs1112 and rs1117 (section II., no interaction), the LRT-based *P*-values strongly depended on the order of the markers in the regression model (`affection ~ rs1117+rs1112`: $p_{\text{rs1117}}=5.547\text{e-}07$ / $p_{\text{rs1112}}=1.193\text{e-}03$; `affection ~ rs1112+rs1117`: $p_{\text{rs1112}}=5.438\text{e-}09$ / $p_{\text{rs1117}}=0.21$). Do you have an explanation?

Both markers are correlated, i.e. they show allelic association (LD), and represent the *same* phenotypic association signal at the locus (remember that association analysis usually pursues an indirect approach). One of the SNPs, namely rs1112, is more strongly correlated with the causative variant than rs1117. The likelihood-ratio test (LRT) compares the following models: `affection~SNP1` vs. `affection~SNP1+SNP2`. If the marker rs1117 is included in the model first as SNP1, it already contains *some but not all* information on the phenotypic association at the locus. Inclusion of rs1112 as SNP2 still contributes significant additional information. However, if marker rs1112 is included first as SNP1, it already contains *all* information available from the data set on the association of the locus. In this case, inclusion of marker rs1117 as SNP2 cannot contribute any further information and is, thus, tested as being insignificant.

Genome-Wide Association Exercise

Association Analysis Controlling for Population Substructure

Copyrighted © 2022 Merry-Lynn N. McDonald, Isabelle Schrauwen & Suzanne M. Leal

1. Population Stratification and Association Testing

The dataset from part I of this exercise which you performed data quality control (QC) on was obtained from HapMap Phase III data. It contains CEU founders (Caucasians from Utah), MEX founders (Mexicans from Los Angeles) and TSI (Tuscans from Italy). The CEU pedigree identifiers begin with only numbers e.g., 1347, the MEX pedigree identifies all start with M e.g., M017 and the TSI pedigree identifiers all start with NA e.g., NA0217. Before we start testing for association, we want to know if there are outliers. Even after removing the outliers when association analysis is performed population substructure and admixture may need to be controlled. If not, we risk observing an association, which is due to a difference in genotype frequencies in cases and controls, because of population substructure/admixture and not because of linkage disequilibrium (LD) between tagSNP(s) and the functional variant(s). We are going to use multidimensional scaling (MDS) and principal components analysis (PCA) within the PLINK software to generate 10 components. **Disclaimer: You usually should not analyze data from European-Americans, Mexican-Americans and Italians together even if you control for population stratification. They can be analyzed separately, and the data combined using meta-analysis.**

Note: For a GWAS study instead of this toy study, you will have a denser set of markers of which some will be in LD. You should first prune your SNPs to obtain a subset in linkage equilibrium/weak LD ($R^2 < 0.5$) prior to performing MDS or PCA analysis on the data. Although for association analysis is performed on the entire data set will be analyzed only this a subset of SNPs which are not in LD will be used to construct PCA and MDS components. For more information on how to do this in PLINK see <https://www.cog-genomics.org/plink/1.9/ld>.

```
plink --file GWAS_clean4 --genome --cluster --mds-plot 10
```

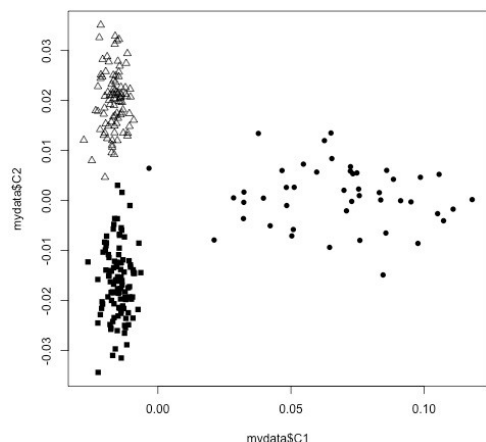
This command outputs the file **plink.mds** that contains the subject IDs and values for the 10 components we just generated. There is another file in your folder called **mds_components.txt**. This file is identical to your **plink.mds** file with the exception that a group column which codes CEU individuals as 1, MEX individuals as 2 and TSI individuals as 3. This is done so when we plot the MDS components in R you can see which group the points belong to and judge how well does the data cluster, e.g., are there outliers. The following commands will generate a jpeg image file containing the mds plot (filename=mds.jpeg) in your current working directory. Open R and use the following command:

```
mydata = read.table("mds_components.txt", header=T)
```

```
mydata$pch[mydata$Group==1 ] <-15  
mydata$pch[mydata$Group==2 ] <-16  
mydata$pch[mydata$Group==3 ] <-2
```

```
jpeg("mds.jpeg", height=500, width=500)  
plot(mydata$C1, mydata$C2 ,pch=mydata$pch)  
dev.off()
```

Visualizing population structure using MDS is useful for identifying subpopulations, population stratification and systematic genotyping or sequencing errors, and can also be used to detect individual outliers that may need to be removed, e.g. European-Americans included in a study of African-Americans. MDS coordinates help with visualizing genetic distances and population substructure.



PLINK also offers another dimension reduction, `--pca`, for PCA, the PC components which can also be used for visualizing data to detect outliers in the same manner which was performed using MDS. Additionally, covariates either from either MDS or PCA can be used in a regression model to aid in correcting for population substructure and admixture.

We will now continue performing the analysis using PLINK but will use PCA instead of MDS. We will generate PCs and determine how many PC covariates should be included in the regression model. When SNPs are tested for an association with a trait analysis can be

performed, first by including no PC components, then one PC component and then two PC components and so on. Please note that as each PC component is added all the SNPs are analyzed, e.g. a complete GWAS is performed. Examining λ can aid in determining how many PC components should be included in the analysis. If there is no population stratification or other biases, then λ should equal 1 or ~ 1 . We will use λ to determine how many PC components from our analysis will be added to the logistic regression model. First, estimate λ without adjusting for any PC components:

```
plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --logistic --adjust -
-out unadj
```

Generated the first 10 PCA values:

```
plink --file GWAS_clean4 --genome --cluster --pca 10 header
```

Eigenvectors are written to `plink.eigenvec`, and top eigenvalues are written to `plink.eigenval`. The 'header' modifier adds a header line to the `.eigenvec` file(s).

And then find out what λ is when we adjust for the first component:

```
plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --covar
plink.eigenvec --covar-name PC1 --logistic --adjust --out PC1
```

And the first and second components:

```
plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --covar
plink.eigenvec --covar-name PC1-PC2 --logistic --adjust --out PC1-PC2
```

and so forth for all 10 components in the `.log` file completing the table:

Table 1											
	Un- adjusted	PC 1	PC 1-2	PC 1-3	PC 1-4	PC 1-5	PC 1-6	PC 1-7	PC 1-8	PC 1-9	PC1- 10
λ											

The number closest to 1.0, with the least number of PC components, would be the best for adjusting without overfitting and introducing unnecessary noise. You can check your table against the one provided in the answers section.

Go to the **assoc.logistic file that corresponds to that number of components** and make a note of how you named the .assoc.logistic file for it and when you did not adjust for any components. Then go back to the R program to load the results and create a jpeg image file containing QQ plots for the adjusted and unadjusted results (using a modified script from <http://www.broad.mit.edu/node/555>) as follows:

```
broadqq <-function(pvals, title)
{
  observed <- sort(pvals)
  lobs <- -(log10(observed))

  expected <- c(1:length(observed))
  lexp <- -(log10(expected / (length(expected)+1)))

  plot(c(0,7), c(0,7), col="red", lwd=3, type="l", xlab="Expected (-logP)", ylab="Observed (-logP)",
  xlim=c(0,max(lobs)), ylim=c(0,max(lobs)), las=1, xaxs="i", yaxs="i", bty="l", main = title)
  points(lexp, lobs, pch=23, cex=.4, bg="black") }

jpeg("qqplot_compare.jpeg", height=1000, width=500)
par(mfrow=c(2,1))
aff_unadj<-read.table("unadj.assoc.logistic", header=TRUE)
aff_unadj.add.p<-aff_unadj[aff_unadj$TEST=="ADD"),]$P
broadqq(aff_unadj.add.p, "Some Trait Unadjusted")
aff_C1C2<-read.table("PC1-PC2.assoc.logistic", header=TRUE)
aff_C1C2.add.p<-aff_C1C2[aff_C1C2$TEST=="ADD"),]$P
broadqq(aff_C1C2.add.p, "Some Trait Adjusted for PC1 and PC2")
dev.off()
```

Now look for SNPs with genome-wide significance using the following R commands:

```
gws_unadj = aff_unadj[which(aff_unadj$P < 0.0000001),]
gws_unadj
gws_adjusted = aff_C1C2[which(aff_C1C2$P < 0.0000001),]
gws_adjusted
```

Note: These are the uncorrected p-values for multiple testing. The p-values which have been corrected using various multiple testing methods can be found in the .adjusted file.

A common question when you have a finding with genome-wide significance in a GWAS is “Is the SNP in a known gene?” One way to look this information up is annotate variants in batch (please look at the annotating exercise for more information). You can do this using the Ensembl Variant Predictor. Go to the website:

http://grch37.ensembl.org/Homo_sapiens/Tools/VEP (GRCh37 version)

Type the rs number(s) of the SNP(s) with genome-wide significance in “Either paste data”, leave all options default and press run. In a few minutes you can view the results of your query.

Question 1: Did this study have a finding with genome-wide significance after adjusting for population substructure? Did you notice any difference in the p-values before and after adjustment for substructure? How many PC components should you include in the regression model. Please also, complete the tables below.

Table 2. SNPS with genome-wide significance unadjusted for substructure:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P

Table 3. SNPs with genome-wide significance adjusted for components 1 and 2:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P

Question 2: Why would you not want to include in your analysis individuals from different ethnic backgrounds even if you control for population substructure?

Question 3. Are any SNPs with genome-wide significance in known genes?

Answers and Output

Table 1

	Un- adjusted	PC1	PC1- 2	PC1- 3	PC1- 4	PC1- 5	PC1- 6	PC1- 7	PC1- 8	PC1- 9	PC1- 10
lambda	1.121	1.085	1.026	1.033	1.040	1.050	1.043	1.021	1.036	1.043	1.051

Answer to Question 1:

Question 1:

Did this study have a finding with genome-wide significance after adjusting for population substructure? How many PC components should you include in the regression model. Did you notice any difference in the p-values before and after adjustment for substructure?

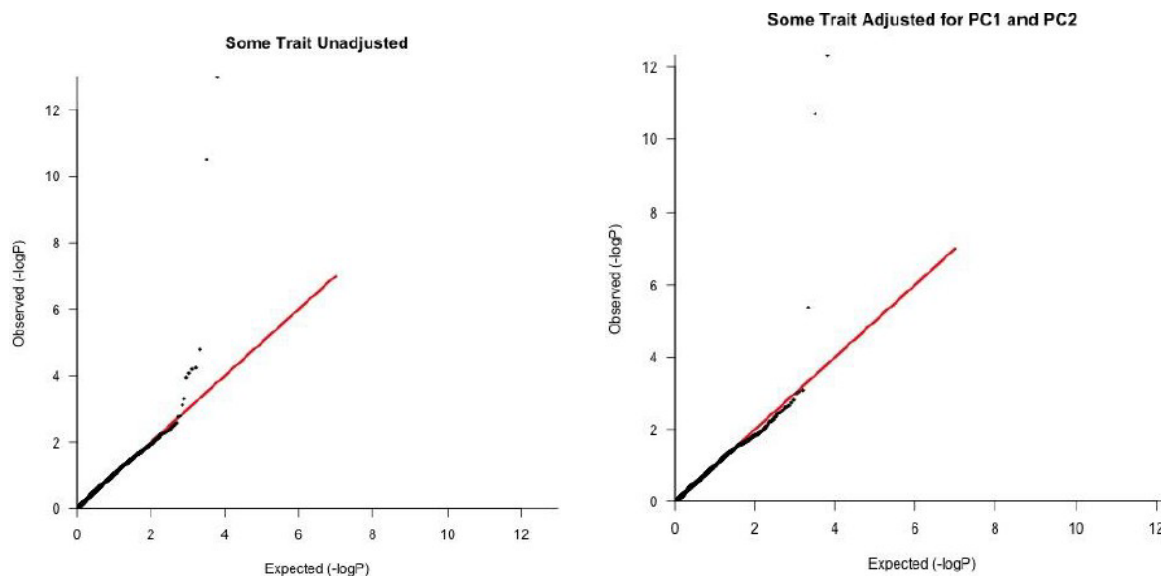
Yes, see tables below. It is best to include two PC components in the analysis, however the lambda is still inflated. Since we are analyzing three unique populations inclusion of PCs did not adequately control for substructure. If you compare the QQ plots below you can see that for this dataset the most significant SNPs were changed minimally when we adjusted for substructure but some of the moderately significant SNPs became less significant after adjustment. However, in some situations the p-values can become smaller.

Table 2. SNPS with genome-wide significance unadjusted for substructure:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
8	rs4571722	60326734	T	ADD	242	0.04126	-7.436	1.04E-13
4	rs10008252	179853616	G	ADD	244	0.1665	-6.639	3.16E-11

Table 3. SNPs with genome-wide significance adjusted for components 1 and 2:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
8	rs4571722	60326734	T	ADD	242	0.04382	-7.237	4.59E-13
4	rs10008252	179853616	G	ADD	244	0.13070	-6.707	1.99E-11



Question 2: Why would you not want to include in your analysis individuals from different ethnic backgrounds even if you control for population substructure?

Firstly, you may not be able to adequately control for population substructure. Secondly, even if within the different populations the same genes are involved, for common variants LD structure can vary between populations, e.g., the tagSNPs in the different populations can have different allele frequencies, therefore the functional variant will not be tagged equally well in all populations and power can be reduced. It is also possible that different variants are associated, but for common variants, which are very old, usually this is not the cause. If a study involves individuals of different ancestry analysis can be performed separately and the results can be combined via meta-analysis. Studying individuals of different ancestry can be highly beneficial to fine map loci.

Question 3: Are any SNPs with genome-wide significance in known genes?

No, both rs457122 and rs10008252 are intergenic/intronic.

Association Analysis of Sequence Data using Variant Association Tools (VAT) for Complex Traits

Copyright (c) 2022 - Gao Wang, Biao Li, Diana Cornejo Sánchez & Suzanne M. Leal

PURPOSE

Variant Association Tools [VAT, Wang et al (2014)] [1] was developed to perform quality control and association analysis of sequence data. It can also be used to analyze genotype data, e.g. exome chip data and imputed data. The software incorporates many rare variant association methods which include but not limited to Combined Multivariate Collapsing (CMC) [2], Burden of Rare Variants (BRV) [3], Weighted Sum Statistic (WSS) [4], Kernel Based Adaptive Cluster (KBAC) [5], Variable Threshold (VT) [6] and Sequence Kernel Association Test (SKAT) [7].

VAT inherits the intuitive command-line interface of Variant Tools (VTools) [8] with re-design and implementation of its infrastructure to accommodate the scale of dataset generated from current sequencing efforts on large populations. Features of VAT are implemented into VTools subcommand system.

RESOURCES

A list of all commands that are used in this exercise can be found at

<https://statgenetics.github.io/statgen-courses/notebooks/VAT.html>

Basic concepts to handle sequence data using vtools can be found at:

<http://varianttools.sourceforge.net/Main/Concepts>

VAT Software documentation

<http://varianttools.sourceforge.net/Main/Documentation>

Genotype data

Exome genotype data was downloaded from the 1000 Genomes pilot data July 2010 release for both the CEU and YRI populations. Only the autosomes are contained in the datasets accompanying this exercise.

The data sets {CEU.exon.2010 03.genotypes.vcf.gz, YRI.exon.2010 03.genotypes.vcf.gz} are available from:

ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/release/2010_07/exon/snps

Phenotype data

To demonstrate the association analysis, we simulated a quantitative trait phenotype (BMI). Please note that these phenotypes are NOT from the 1000 genome project.

Computation resources

Due to the nature of next-generation sequencing data, a reasonably powerful machine with high speed internet connection is needed to use this tool for real-world applications. For this reason, in this tutorial we will use a small demo dataset to demonstrate association analysis.

1 Data Quality Control, Annotation and Variant/sample Selection - Part I

1.1 Getting started

Check the available subcommands by typing:

```
vtools -h
```

Subcommand system is used for various data manipulation tasks (to check details of each subcommand use `vtools name -h`). This tutorial is mission oriented and focuses on a subset of the commands that are relevant to variant-phenotype association analysis, rather than introducing them systematically. For additional functionality, please refer to documentation and tutorials online.

Initialize a project

```
vtools init VATDemo
```

OUTPUT

```
INFO: variant tools 3.0.9 : Copyright (c) 2011 - 2016 Bo Peng
INFO: Please visit https://github.com/vatlab/varianttools for more information.
INFO: Creating a new project VATDemo
```

Command `vtools init` creates a new project in the current directory. A directory can only have one project. After a project is created, subsequent `vtools` calls will automatically load the project in the current directory. Working from outside of a project directory is not allowed.

Import variant and genotype data

Import all vcf files under the current directory:

```
vtools import *.vcf.gz --var_info DP filter --geno_info DP_geno --build hg18 -j1
```

OUTPUT

```
INFO: Importing variants from CEU.exon.2010_03.genotypes.vcf.gz (1/2)
CEU.exon.2010_03.genotypes.vcf.gz: 100% [=====] 4,306 3.1K/s in 00:00:01
INFO: 3,489 new variants (3,489 SNVs) from 3,500 lines are imported.
Importing genotypes: 100% [=====] 3,489 10.7K/s in 00:00:00
INFO: Importing variants from YRI.exon.2010_03.genotypes.vcf.gz (2/2)
YRI.exon.2010_03.genotypes.vcf.gz: 100% [=====] 5,967 10.8K/s in 00:00:00
INFO: 3,498 new variants (5,175 SNVs) from 5,186 lines are imported.
Importing genotypes: 100% [=====] 6,987 22.7K/s in 00:00:00
```

Command `vtools import` imports variants, sample genotypes and related information fields. The imported variants are saved to the master variant table for the project, along with their information fields.

The command above imports two vcf files sequentially into an empty `vtools` project. The second INFO message in the screen output shows that 3,489 variant sites are imported from the first vcf file, where 3,489 new means that all of them are new because prior to importing the first vcf the project was empty so there was 0 site. The fourth INFO message tells that 5,175 variant sites are imported from the second vcf file, but only 3,498 of them are new (which are not seen in the existing 3,489) because prior to importing the second vcf there were already 3,489 existing variant sites from first vcf.

Thus, $5,175 - 3,498 = 1,677$ variant sites are overlapped sites between first and second vcfs. More details about `vtools import` command can be found at <http://varianttools.sourceforge.net/Vtools/Import>

Since the input VCF file uses hg18 as the reference genome while most modern annotation data sources are hg19-based, we need to *liftover* our project using hg19 in order to use various annotation sources in the analysis. Vtools provides a command which is based on the tool of UCSC liftOver to map the variants from existing reference genome to an alternative build. More details about `vtools liftover` command can be found at <http://varianttools.sourceforge.net/Vtools/Liftover>

```
vtools liftover hg19 --flip
```

OUTPUT

```
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [=====] 6,987
333.2K/s in 00:00:00
INFO: Running UCSC liftOver tool
INFO: Flipping primary and alternative reference genome
Updating table variant: 100% [=====] 6,987
45.1K/s in 00:00:00
```

Import phenotype data

The aim of the association test is to find variants that modulate the phenotype BMI. We simulated BMI values for each of the individuals. The phenotype file must be in plain text format with sample names matching the sample IDs in the vcf file(s):

```
head phenotypes.csv
```

```
_____ .phenotypes.csv _____
```

```
sample_name,panel,SEX,BMI
NA06984,ILLUMINA,1,36.353
NA06985,NA,2,21.415
NA06986,ABI_SOLID+ILLUMINA,1,26.898
NA06989,ILLUMINA,2,25.015
NA06994,ABI_SOLID+ILLUMINA,1,23.858
NA07000,ABI_SOLID+ILLUMINA,2,36.226
NA07037,ILLUMINA,1,32.513
NA07048,ILLUMINA,2,17.57
NA07051,ILLUMINA,1,37.142
```

The phenotype file includes information for every individual, the sample name, sequencing panel, sex and BMI. To import the phenotype data:

```
vtools phenotype --from_file phenotypes.csv --delimiter ","
```

```
_____ OUTPUT _____
```

```
INFO: Adding phenotype panel of type VARCHAR(24)
INFO: Adding phenotype SEX of type INT
INFO: Adding phenotype BMI of type FLOAT
INFO: 3 field (3 new, 0 existing) phenotypes of 202 samples are updated.
```

Unlike `vtools import`, this command imports/adds properties to samples rather than to variants. More details about `vtools phenotype` command can be found at <http://varianttools.sourceforge.net/Vtools/Phenotype>

View imported data

Summary information for the project can be viewed anytime using the command `vtools show`, which displays various project and system information. More details about `vtools show` can be found at <http://varianttools.sourceforge.net/Vtools/Show>. Some useful data summary commands are:

```
vtools show project
vtools show tables
vtools show table variant
vtools show samples
vtools show genotypes
vtools show fields
```

1.2 Overview of variant and genotype data

Total number of variants

The number of imported variants may be greater than number of lines in the vcf file, because when a variant has two alternative alleles (e.g. A->T/C) it is treated as two separate variants.

```
vtools select variant --count
```

There are 6987 variants in our test data.

`vtools select table condition action` selects from a variant table `table` a subset of variants satisfying a specified condition, and perform an action of

- creating a new variant table if `--to table` is specified.

- counting the number of variants if `--count` is specified.
- outputting selected variants if `--output` is specified.

The condition should be a SQL expression using one or more fields in a project (displayed in `vtools show fields`). If the condition argument is unspecified, then all variants in the table will be selected. An optional condition `--samples [condition]` can also be used to limit selected variants to specific samples. More details about `vtools select` command can be found at <http://varianttools.sourceforge.net/Vtools/Select>

Genotype Summary

The command `vtools show genotypes` displays the number of genotypes for each sample and names of the available genotype information fields for each sample, e.g. GT - genotypē; DP geno - genotype read depth. Such information is useful for the calculation of summary statistics of genotypes (e.g. depth of coverage).

```
vtools show genotypes > GenotypeSummary.txt
head GenotypeSummary.txt
```

sample name	Filename	num genotypes	sample genotype fields
NA06984	CEU.exon.2010 03.genotypes.vcf.gz	3162	GT,DP geno -
NA06985	CEU.exon.2010 03.genotypes.vcf.gz	3144	GT,DP geno -
NA06986	CEU.exon.2010 03.genotypes.vcf.gz	3437	GT,DP geno -
NA06989	CEU.exon.2010 03.genotypes.vcf.gz	3130	GT,DP geno -
NA06994	CEU.exon.2010 03.genotypes.vcf.gz	3002	GT,DP geno -
NA07000	CEU.exon.2010 03.genotypes.vcf.gz	3388	GT,DP geno -
NA07037	CEU.exon.2010 03.genotypes.vcf.gz	3374	GT,DP geno -
NA07048	CEU.exon.2010 03.genotypes.vcf.gz	3373	GT,DP geno -
NA07051	CEU.exon.2010 03.genotypes.vcf.gz	3451	GT,DP geno -

Variant Quality Overview

The following command calculates summary statistics on the variant site depth of coverage (DP). Below is the command to calculate depth of coverage information for all variant sites.

```
vtools output variant "max(DP) " "min(DP) " "avg(DP) " "stdev(DP) " "lower_quartile(DP) "
"upper_quartile(DP) " --header
```

max DP -	min DP -	avg DP -	stdev DP -	lower quartile DP -	upper quartile DP -
25490	13	6815.77028768	3434.28040091	4301	9143

In the test data, the maximum DP for variant sites is 25490, minimum DP 13, average DP about 6815, standard deviation of DP about 3434, lower quartile of DP 4301 and upper quartile of DP 9143.

The same syntax can be applied to other variant information or annotation information fields. The command `vtools output name` of variant table outputs properties of variants in a specified variant table. The properties include fields from annotation databases and variant tables, basically fields outputted from command `vtools show fields`, and SQL-supported functions and expressions. There are several freely available SQL resources on the web to learn more about SQL functions and expressions.

It is also possible to view variant level summary statistic for variants satisfying certain filtering criteria using `vtools select-name` of variant table command, for example to count only variants having passed all quality filters:

```
vtools select variant "filter='PASS'" --count
```

All 6987 variants have passed the quality filters. To combine variant filtering and summary statistics:

```
vtools select variant "filter='PASS'" -o "max(DP) " "min(DP) " "avg(DP) " "stdev(DP) "
"lower_quartile(DP) " "upper_quartile(DP) " --header
```

The output information of command above will be the same as the previous `vtools output` command, since all variants have passed quality filter.

1.3 Data exploration

Variant level summaries

The command below will calculate:

- **total**: Total number of genotypes (GT) for a variant
- **num**: Total number of alternative alleles across all samples
- **het**: Total number of heterozygote genotypes 1/0
- **hom**: Total number of homozygote genotypes 1/1
- **other**: Total number of double-homozygotes 1/2
- **min/max/meanDP**: Summaries for depth of coverage and genotype quality across samples
- **maf**: Minor allele frequency
- Add calculated variant level statistics to fields, which can be shown by commands `vtools show fields` and `vtools show table variant`

```
vtools update variant --from_stat 'total=#(GT)' 'num=#(alt)' 'het=#(het)' 'hom=#(hom)'  
'other=#(other)' 'minDP=min(DP_geno)' 'maxDP=max(DP_geno)' 'meanDP=avg(DP_geno)' 'maf=maf()'
```

OUTPUT

```
INFO: Reading genotype info for processing...  
INFO: Adding variant info field num with type INT  
INFO: Adding variant info field hom with type INT  
INFO: Adding variant info field het with type INT  
INFO: Adding variant info field other with type INT  
INFO: Adding variant info field total with type INT  
INFO: Adding variant info field maf with type FLOAT  
INFO: Adding variant info field minDP with type INT  
INFO: Adding variant info field maxDP with type INT  
INFO: Adding variant info field meanDP with type FLOAT
```

```
Updating variant: 100% [=====] 6,987 42.5K/s in 00:00:00
```

```
vtools show fields  
vtools show table variant
```

Command `vtools update` updates variant info fields (and to a lesser extend genotype info fields) by adding more fields or updating values at existing fields. It does not add any new variants or genotypes, and does not change existing variants, samples, or genotypes. Using three parameters `--from file`, `--from stat`, and `--set`, variant information fields could be updated from external file, sample genotypes, and existing fields. More details about `vtools update` command can be found at <http://varianttools.sourceforge.net/Vtools/Update>

Summaries for different genotype depth (GD) and genotype quality (GQ) filters

The `--genotypes CONDITION` option restricts calculation to genotypes satisfying a given condition. Later we will remove individual genotypes by `DP geno` filters. The command below will calculate summary statistics genotypes of all samples per variant site. It can assist us in determining filtering criteria for genotype call quality.

```
vtools update variant --from_stat 'totalGD10=#(GT)' 'numGD10=#(alt)' 'hetGD10=#(het)'  
'homGD10=#(hom)' 'otherGD10=#(other)' 'mafGD10=maf()' --genotypes "DP_geno > 10"
```

OUTPUT

```
INFO: Reading genotype info for processing...  
INFO: Adding variant info field numGD10 with type INT  
INFO: Adding variant info field homGD10 with type INT  
INFO: Adding variant info field hetGD10 with type INT  
INFO: Adding variant info field otherGD10 with type INT  
INFO: Adding variant info field totalGD10 with type INT  
INFO: Adding variant info field mafGD10 with type FLOAT
```

```
Updating variant: 100% [=====] 6,987 52.1K/s in 00:00:00
```

```
vtools show fields  
vtools show table variant
```

You will notice the change in genotype counts when applying the filter on genotype depth of coverage and only retaining those genotypes with a read depth greater than 10X. There are now 6987 variant sites after filtering on

DP geno>10. Note that some variant sites will become monomorphic after removing genotypes due to low read depth.

Minor allele frequencies (MAFs)

In previous steps, we calculated MAFs for each variant site before and after filtering on genotype read depth. Below is a summary of the results:

```
vttools output variant chr pos maf mafGD10 --header --limit 20
```

OUTPUT			
chr	pos	Maf	mafGD10
1	1105366	0.0350877192982	0.0512820512821
1	1105411	0.00943396226415	0.0128205128205
1	1108138	0.192307692308	0.18023255814
1	1110240	0.00561797752809	0.0
1	1110294	0.228125	0.242307692308
1	3537996	0.12012987013	0.152173913043
1	3538692	0.0410256410256	0.0432098765432
1	3541597	0.00561797752809	0.00617283950617
1	3541652	0.0444444444444	0.0533333333333
1	3545211	0.00561797752809	0.00581395348837
...			

Adding "> filename.txt" at the end of the above command will write the output to a file.

Next, we examine population specific MAFs. Our data is imported from two files, a CEU dataset (90 samples) and an YRI dataset (112 samples). To calculate allele frequency for each population, let us first assign an additional RACE phenotype (0 for YRI samples and 1 for CEU samples):

```
vttools phenotype--set "RACE=0" --samples "filename like 'YRI%'"
vttools phenotype--set "RACE=1" --samples "filename like 'CEU%'"
vttools show samples --limit 10
```

OUTPUT					
sample_name	filename	panel	SEX	BMI	RACE
NA06984	CEU.exon...notypes.vcf.gz	ILLUMINA	1	36.353	1
NA06985	CEU.exon...notypes.vcf.gz	.	2	21.415	1
NA06986	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	26.898	1
NA06989	CEU.exon...notypes.vcf.gz	ILLUMINA	2	25.015	1
NA06994	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	23.858	1
NA07000	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	2	36.226	1
NA07037	CEU.exon...notypes.vcf.gz	ILLUMINA	1	32.513	1
NA07048	CEU.exon...notypes.vcf.gz	ILLUMINA	2	17.57	1
NA07051	CEU.exon...notypes.vcf.gz	ILLUMINA	1	37.142	1
NA07346	CEU.exon...notypes.vcf.gz	. 2 30.978 1 (192 records omitted)			

Population specific MAF calculations will be performed using those genotypes that passed the read depth filter (DP geno>10)

```
vttools update variant --from_stat 'CEU_mafGD10=maf()' --genotypes 'DP_geno>10' --samples "RACE=1"
vttools update variant --from_stat 'YRI_mafGD10=maf()' --genotypes 'DP_geno>10' --samples "RACE=0"
vttools output variant chr pos mafGD10 CEU_mafGD10 YRI_mafGD10 --header --limit 10
```

OUTPUT				
chr	Pos	mafGD10	CEU_mafGD10	YRI_mafGD10
1	1105366	0.0512820512821	0.0512820512821	0.0
1	1105411	0.0128205128205	0.0128205128205	0.0
1	1108138	0.18023255814	0.0212765957447	0.371794871795
1	1110240	0.0	0.0	0.0
1	1110294	0.242307692308	0.025	0.428571428571
1	3537996	0.152173913043	0.170454545455	0.135416666667

1	3538692	0.0432098765432	0.0833333333333	0.00595238095238
1	3541597	0.00617283950617	0.00617283950617	0.0
1	3541652	0.0533333333333	0.0533333333333	0.0
1	3545211	0.00581395348837	0.00581395348837	0.0

You will observe zero values because some variant sites are monomorphic or they are population specific.

Sample level genotype summaries

Similar operations could be performed on a sample level instead of on a variant level. More details about obtaining genotype level summary information using `vtools phenotype --from stat` can be found at <http://varianttools.sourceforge.net/Vtools/Phenotype>

```
vtools phenotype --from_stat 'CEU_totalGD10=#(GT)' 'CEU_numGD10=#(alt)' --genotypes 'DP_geno>10' --samples "RACE=1"
vtools phenotype --from_stat 'YRI_totalGD10=#(GT)' 'YRI_numGD10=#(alt)' --genotypes 'DP_geno>10' --samples "RACE=0"
```

OUTPUT

180 values of 2 phenotypes (2 new, 0 existing) of 90 samples are updated.
224 values of 2 phenotypes (2 new, 0 existing) of 112 samples are updated.

```
vtools phenotype --output sample_nameCEU_totalGD10CEU_numGD10YRI_totalGD10YRI_numGD10 --header
```

OUTPUT

sample_name	CEU_totalGD10	CEU_numGD10	YRI_totalGD10	YRI_numGD10
NA06984	2774	849 NA	NA	NA
NA06985	1944	570 NA	NA	NA
NA06986	3386	1029 NA	NA	NA
NA06989	2659	819 NA	NA	NA
NA06994	1730	486 NA	NA	NA
...				
NA19257	NA	NA 4969	1229	
NA19259	NA	NA 4182	1005	
NA19260	NA	NA 4404	1076	
NA19262	NA	NA 4308	1044	
NA19266	NA	NA 4878	1211	

1.4 Variant Annotation

For rare variant aggregated association tests, we want to focus on analyzing aggregating variants having potential functional contribution to a phenotype. Thus, each variant site needs to be annotated for its functionality. Annotation is performed using variant annotation tools [7] which implements an ANNOVAR pipeline for variant function annotation [9]. More details about the ANNOVAR pipeline can be found at <http://varianttools.sourceforge.net/Pipeline/Annovar>

```
vtools execute ANNOVAR geneanno
```

OUTPUT

```
INFO: Running vtools update variant --from_file cache/annovar_input.variant_function --format ANNOVAR_variant_function
n --var_info region_type, region_name
...
Running vtools update variant --from_file cache/annovar_input.exonic_variant_function --format
ANNOVAR_exonic_variant_function --var_info mut_type, function
...
INFO: Fields mut_type, function of 6,920 variants are updated
```

The following command will output the annotated variant sites to the screen.

```
vtools output variant chr pos ref alt mut_type --limit 20 --header
```

OUTPUT				
chr	pos	ref	alt	mut type
1	1105366	T	C	nonsynonymous SNV
1	1105411	G	A	nonsynonymous SNV
1	1108138	C	T	synonymous SNV
1	1110240	T	A	nonsynonymous SNV
1	1110294	G	A	nonsynonymous SNV
1	3537996	T	C	synonymous SNV
...				

Many more annotation sources are available which are not covered in this tutorial. Please read <http://varianttools.sourceforge.net/Annotation> for annotation databases, and <http://varianttools.sourceforge.net/Pipeline> for annotation pipelines.

1.5 Data Quality Control (QC) and Variant Selection

Ti/Tv ratio evaluations

Before performing any data QC we examine the transition/transversion (Ti/Tv) ratio for all variant sites. Note that here we are obtaining Ti/Tv ratios for the entire sample, Ti/Tv ratios can also be obtained for each sample.

```
vtools_report trans_ratio variant -n num
```

num of transition	num of transversion	ratio
161,637	44,641	3.62082

The command above counts the number of transition and transversion variants and calculates its ratio. More details about `vtools_report trans_ratio` command can be found at <http://varianttools.sourceforge.net/VtoolsReport/TransRatio>

If only genotype calls having depth of coverage greater than 10 are considered:

```
vtools_report trans_ratio variant -n numGD10
```

num of transition	num of transversion	ratio
140,392	38,710	3.62676

We can see that Ti/Tv ratio has increase slightly if low depth of coverage calls are removed. There is only a small change in the Ti/Tv ratio since only a few variant sites become monomorphic and are no longer included in the calculation. In practice Ti/Tv ratios can be used to evaluate which threshold should be used in data QC.

Removal of low quality variant sites

We should not need to remove any variant site based on read depth because all variants passed the quality filter. To demonstrate removal of variant sites, let us

```
remove those with a total read depth {$(\le)} 15.
vtools select variant "DP<15" -t to_remove
vtools show tables
vtools remove variants to_remove -v0
vtools show tables
```

We can see that one variant site has been removed from master variant table. The `vtools remove` command can remove various items from the current project. More details about `vtools remove` command can be found at <http://varianttools.sourceforge.net/Vtools/Remove>. Using a combination of select/remove subcommands low quality variant sites can be easily filtered out. The `vtools show fields`,

vtools show tables, and vtools show table variant commands will allow you to see the new/updated fields and tables you have added/changed to the project.

Filter genotype calls by quality

We have calculated various summary statistics using the command `--genotypes 'CONDITION'` but we have not yet removed genotypes having genotype read depth of coverage lower than 10X. The command below removes these genotypes.

```
vtools remove genotypes "DP_geno<10" -v0
```

Select variants by annotated functionality

To select potentially functional variants for association mapping:

```
vtools select variant "mut_type like 'non%' or mut_type like 'stop%' or region_type='splicing'"
-t v_funcnt
vtools show tables
```

The command above selects variant sites that are either nonsynonymous (by condition `"mut type like 'non%'"`) or stop-gain/stop-loss (by condition `mut type like 'stop%'"`) or alternative splicing (by condition `region-type='splicing'"`)

3367 functional variant sites are selected

2 Association Tests for Quantitative Traits - Part II

2.1 View phenotype data

```
vtools show samples --limit 5
```

OUTPUT					
sample_name	filename	panel	SEX	BMI	...
NA06984	CEU.exon...notypes.vcf.gz	ILLUMINA	1	36.353	...
NA06985	CEU.exon...notypes.vcf.gz	.	2	21.415	...
NA06986	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	26.898	...
NA06989	CEU.exon...notypes.vcf.gz	ILLUMINA	2	25.015	...
NA06994	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	23.858	...

2.2 Analysis plan

We want to carry out the association analysis for CEU and YRI separately. For starters we demonstrate analysis of CEU samples; and the same commands will be applicable for YRI samples. After completing the analysis of CEU samples please use the same commands to analyze the YRI data set. You should not analyze the data from different populations together, once you have the p-values from each analysis, you may perform a meta-analysis.

2.3 Subset data by MAFs

To carry out association tests we need to treat common and rare variants separately. The dataset for our tutorial has very small sample size, but with large sample size it is reasonable to define rare variants as having observed $MAF < 0.01$, and common variants as variants having observed $MAF \geq 0.05$. First, we create variant tables based on calculated alternative allele frequencies for both populations

```
vtools select variant "CEU_mafGD10>=0.05" -t common_ceu
```

```
vtools select v_funcnt "CEU_mafGD10<0.01" -t rare_ceu
```

Notice that for selection of rare variants we only keep those that are annotated as functional (chosen from v_funcnt table). There are 1450 and 604 variant sites selected for $MAF \geq 0.05$ and $MAF < 0.01$, respectively.

2.4 Annotate variants to genes

For gene based rare variant analysis we need annotations that tell us the boundaries of genes. We use the refGene annotation database for this purpose.

```
vtools use refGene
```

OUTPUT

```
INFO: Downloading annotation database annoDB/refGene-hg19_20130904.ann
INFO: Downloading annotation database from annoDB/refGene-hg19_20130904.DB.gz refGene-hg19_20130904.DB.gz:
100% [=====] 8,056,345.0
411.6K/s in 00:00:19
INFO: Using annotation DB refGene as refGene in project ceu.
INFO: Known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference
sequences collection (RefSeq).
```

```
vtools show annotation refGene
```

OUTPUT

```
Annotation database refGene (version hg19_20130904)
Description:      Known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference seq
uences collection (RefSeq).
Database type:    range
Reference genome hg19: chr, txStart, txEnd
  name (char)      Gene name
  chr (char)
  strand (char)     which DNA strand contains the observed alleles
  txStart (int)     Transcription start position (1-based)
  txEnd (int)       Transcription end position
  cdsStart (int)    Coding region start (1-based)
  cdsEnd (int)      Coding region end
  exonCount (int)   Number of exons
  exonStarts (char) Starting point of exons (adjusted to 1-based positions)
  exonEnds (char)   Ending point of exons
  score (int)       Score
  name2 (char)      Alternative name
  cdsStartStat (char) cds start stat, can be 'non', 'unk', 'incompl', and 'cml'
  cdsEndStat (char) cds end stat, can be 'non', 'unk', 'incompl', and 'cml'
```

The names of genes are contained in the refGene.name2 field. The vtools use command, attaches an annotation database to the project, effectively incorporating one or more attributes available to variants in the project. More details about vtools use command can be found at <http://varianttools.sourceforge.net/Vtools/Use>

2.5 Association testing of common/rare variants

The association test program VAT is currently under development and is temporarily implemented as the vtools associate subcommand. To list available association test options

```
vtools associate -h
vtools show tests
vtools show test LinRegBurden
```

Note that we use the quantitative trait BMI as the phenotype, and we will account for “SEX” as a covariate in the regression framework. More details about `vtools associate` command can be found at <http://varianttools.sourceforge.net/Vtools/Associate>

Analysis of common variants

By default, the program will perform single variant tests using a simple linear model, and the Wald test statistic will be evaluated for p-values:

```
vtools associate common_ceu BMI --covariate SEX -m "LinRegBurden --
alternative 2" -j1 --to_db EA_CV > EA_CV.asso.res
```

OUTPUT

```
INFO: 90 samples are found
INFO: 1450 groups are found
Loading genotypes: 100% [=====] 90 56.7/s in 00:00:01
Testing for association: 100% [=====] 1,450/5 684.5/s in 00:00:02
INFO: Association tests on 1450 groups have completed. 5 failed.
INFO: Using annotation DB EA_CV as EA_CV in project ceu.
INFO: Annotation database used to record results of association tests. Created on Fri, 25 Mar 2016 17:45:52
INFO: 1450 out of 3484 variant.chr, variant.pos are annotated through annotation database EA_CV
```



Note

Option `-j1` specifies that 1 CPU core be used for association testing. You may use larger number of jobs for real world data analysis, e.g., use `-j16` if your computational resources has 16 CPU cores available. Linux command `cat /proc/cpuinfo` shows the number of cores and other information related to the CPU on your computer.

Association tests on 1450 groups have completed. 5 failed.

The following command displays error messages about the failed tests. In each case, the sample size was too small to perform the regression analysis.

```
grep -i error *.log
```

OUTPUT

```
2016-03-25 12:45:57,373: DEBUG: An ERROR has occurred in process 0 while processing '6:30018583':
Sample size too small (2) to be analyzed for '6:30018583'.
2016-03-25 12:45:57,378: DEBUG: An ERROR has occurred in process 0 while processing '6:30018721':
Sample size too small (2) to be analyzed for '6:30018721'.
2016-03-25 12:45:57,574: DEBUG: An ERROR has occurred in process 0 while processing '7:148552665':
Sample size too small (2) to be analyzed for '7:148552665'.
2016-03-25 12:45:57,662: DEBUG: An ERROR has occurred in process 0 while processing '8:145718728':
Sample size too small (4) to be analyzed for '8:145718728'.
2016-03-25 12:45:57,669: DEBUG: An ERROR has occurred in process 0 while processing '9:205057': Sample
size too small(4) to be analyzed for '9:205057'.
```

A summary from the association test is written to the file `EA_CV.asso.res`. The first column indicates the variant chromosome and base pair position so that you may follow up on the top signals using various annotation sources that we will not introduce in this tutorial. The result will be automatically built into annotation database if `--to_db` option is specified.

You may view the summary using the `less` command

```
less EA_CV.asso.res
```

To sort the results by p-value and output the first 10 lines of the file use the command:

```
sort -g -k7 EA_CV.asso.res | head
```

If you obtain significant p-values be sure to also observe the accompanying sample size. Significant p-values from too small of a sample size may not be results you can trust.

Also, depending on your phenotype you may have to add additional covariates to your analysis. VAT allows you to test many different models for the various phenotypes and covariates. P-values for covariates are also reported.

Similar to using an annotation database, you can use the results from the association test to annotate the project and follow up variants of interest, for example:

```
vtools show fields
```

association analysis result columns	
Field name	Description
EA_CV.variant_chr	
EA_CV.variant_pos	
EA_CV.sample_size_LinRegBurden	
EA_CV.beta_x_LinRegBurden	
EA_CV.pvalue_LinRegBurden	
EA_CV.wald_x_LinRegBurden	
EA_CV.beta_2_LinRegBurden	
EA_CV.beta_2_pvalue_LinRegBurden	
EA_CV.wald_2_LinRegBurden	
variant_chr	
variant_pos	
sample_size	
test statistic	In the context of regression, this is estimate of effect size for x p-value
Wald statistic for x (beta_x/SE(beta_x))	
estimate of beta for covariate 2	
p-value for covariate 2	
Wald statistic for covariate 2	

You see additional annotation fields starting with EA_CV, the name of the annotation database you just created from association test (if you used the `--to db` option mentioned above). You can use them to easily select/output variants of interest. More details about outputting annotation fields for significant findings can be found at <http://varianttools.sourceforge.net/Vtools/Output>

Burden test for rare variants (BRV)

BRV method uses the count of rare variants in given genetic region for association analysis, regardless of the region length.

We use the `-g` option and use the `'refGene.name2'` field to define the boundaries of a gene. By default, the test is a linear regression using aggregated counts of variants in a gene region as the regressor.

```
vtools associate rare_ceu BMI --covariate SEX -m "LinRegBurden --alternative 2" -g refGene.name2 -j1 --to_db EA_RV > EA_RV.asso.res
```

OUTPUT
INFO: 90 samples are found
INFO: 254 groups are found
Loading genotypes: 100% [=====] 90 48.6/s in 00:00:01
Testing for association: 100% [=====] 254/20 685.4/s in 00:00:00
INFO: Association tests on 254 groups have completed. 20 failed.
INFO: Using annotation DB EA_RV as EA_RV in project ceu.
INFO: Annotation database used to record results of association tests. Created on Fri, 25 Mar 2016 17:47:26
INFO: 254 out of 25360 refGene.refGene.name2 are annotated through annotation database EA_RV

Association tests on 254groups have completed. 20 failed. To view failed tests:

```
grep -i error *.log | tail -10
```

OUTPUT

```
2016-03-25 12:49:49,553: DEBUG: An ERROR has occurred in process 0 while processing 'ABCC1': No variant found in genotype data for 'ABCC1'.
2016-03-25 12:49:49,620: DEBUG: An ERROR has occurred in process 0 while processing 'ANO9': No variant found in genotype data for 'ANO9'.
2016-03-25 12:49:49,781: DEBUG: An ERROR has occurred in process 0 while processing 'C10orf71': No variant found in genotype data for 'C10orf71'.
2016-03-25 12:49:49,875: DEBUG: An ERROR has occurred in process 0 while processing 'CCDC127': No variant found in genotype data for 'CCDC127'.
2016-03-25 12:49:50,313: DEBUG: An ERROR has occurred in process 0 while processing 'FBXL13': No variant found in genotype data for 'FBXL13'.
...
```

The output file is `EA_RV.asso.res`. The first column is the gene name, with corresponding p-values in the sixth column for the entire gene.

```
less EA_RV.asso.res
```

You can also sort these results by p-value using command:

```
sort -g -k6 EA_RV.asso.res | head
```

Variable thresholds test for rare variants (VT)

The variable thresholds (VT) method will carry out multiple testing in the same gene region using groups of variants based on observed variant allele frequencies. This test will maximize over statistics thus obtain a final test statistic, and calculate the empirical p-value so that multiple comparisons are adjusted for correctly.

We will use adaptive permutation to obtain empirical p-values. Therefore, to avoid performing too large number of permutations we use a cutoff to limit the number of permutations when the p-value is greater than 0.0005, e.g. not all 100,000 permutations are performed. Generally, even more permutations are used but we limit it to 100,000 to save time for this exercise.

The command using variable thresholds method on our data is:

```
vtools associate rare_ceu BMI --covariate SEX -m "VariableThresholdsQt --alternative 2
-p 100000 \ --adaptive 0.0005" -g refGene.name2 -j1 --to_db EA_RV > EA_RV_VT.asso.res
```

To view test that failed,

```
grep -i error *.log | tail -10
```

To view results,

```
less EA_RV_VT.asso.res
```



Note

The p values you obtained for VT might be slightly different for each run. This is due to the randomness in permutation tests.

Sort and output the lowest p-values using the command:

```
sort -g -k6 EA_RV_VT.asso.res | head
```

Why do some tests fail?

Notice that `vtools associate` command will fail on some association test units. Instances of failure are printed to terminal in red and are recorded in the project log file. Most failures occur due to an association test unit having too few samples or number of variants (for gene based analysis). You should view these error

messages after each association scan is complete, e.g., using the Linux command `grep -i error *.log` and make sure you are informed of why failures occur.

In the variable thresholds analysis above, gene `ABCC1` failed the association test. If we look at this gene more closely we can see which variants are being analyzed by our test:

```
vtools select rare_ceu "refGene.name2='ABCC1'" -o chr pos ref alt CEU_mafGD10 numGD10 mut_type --header
```

chr	Pos	ref	alt	CEU_mafGD10	numGD10	mut type
16	16178858	T	C	0.0	243	nonsynonymous SNV

After applying our QC filters we are left with one variant within the `ABCC1` gene to analyze. Because the MAF for this variant is 0.0 there are no variants in the gene to analyze so that this gene is ignored. Note that all individuals are homozygous for the alternative allele for this variant site.

QQ and Manhattan plots for association results

The `vtools report plot association` command generates QQ and Manhattan plots from output of `vtools associate` command. More details about `vtools report plot association` can be found at <http://varianttools.sourceforge.net/VtoolsReport/PlotAssociation>

```
vtools_report plot_association qq -o QQRV -b --label_top 2 -f 6 < EA_RV.asso.res
vtools_report plot_association manhattan -o MHRV -b --label_top 5 --color Dark2 --
chrom_prefix None -f 6 < EA_RV.asso.res
```

QQ plots aid in evaluating if there is systematic inflation of test statistics. A common cause of inflation is population structure or batch effects. If you observe significant inflation of test you may consider including MDS components in the association test model.

```
vtools associate rare_ceu BMI --covariate SEX KING_MDS1 KING_MDS2 -m "LinRegBurden --name RVMS2 --alternative 2" -\
g refGene.name2 -j1 --to_db EA_RV > EA_RV_MDS2.asso.res
vtools_report plot_association qq -o QQRV_MDS2 -b -- label_top 2 -f 6 < EA_RV_MDS2.asso.res
```

To visualize the plots copy them to the work directory by typing:

```
$ cp MHRV.pdf /home/jovyan/work
```

```
$ cp QQRV.pdf /home/jovyan/work
```

Now visualize from your computer's home directory

You should not arbitrarily include MDS (or PCA) components in the analysis. Instead put in each MDS component and examine the lambda value, i.e. include MDS component 1 then MDS components 1 and 2, etc. Visualization of the QQ plot is also useful to determine if population substructure/admixture is controlled

2.6 Association analysis of YRI samples

Procedures for YRI sample association analysis is the same as for CEU samples as previously has been described, thus is left as an extra exercise for you to work on your own. Commands to perform analysis for YRI are found below:

```
cd ..
vtools select variant --samples "RACE=0" -t YRI
mkdir -p yri; cd yri
vtools init yri --parent ../ --variants YRI --samples
"RACE=0" --build hgl9 vtools select variant
"YRI_mafGD10>=0.05" -t common_yri vtools select v_func
"YRI_mafGD10<0.01" -t rare_yri
vtools use refGene
vtools associate common_yri BMI --covariate SEX -m "LinRegBurden --alternative 2" -j1 --to_db YA_CV > YA_CV.asso.res
```

```
vtools associate rare_yri BMI --covariate SEX -m "LinRegBurden --alternative 2" -g refGene.name2 -j1 --
to_db YA_RV > YA_RV.asso.res vtools associate rare_yri BMI --covariate SEX -m "VariableThresholdsQt --
alternative 2 -p 100000 \
--adaptive 0.0005" -g refGene.name2 -j1 --to_db YA_RV
> YA_RV_VT.asso.res cd ..
```

2.7 MDS analysis and PC adjustment

This pipeline needs [PLINK 1.9](#) and [KING](#).

```
vtools execute KING
$ cp KING.mds.pdf /home/jovyan/work
```

2.8 Meta-analysis

Here we demonstrate the application of meta-analysis to combine association results from the two populations via vtools report meta analysis. More details about vtools report meta analysis command can be found at

<http://varianttools.sourceforge.net/VtoolsReport/MetaAnalysis->

The input to this command are the association results files generated from previous steps, for example:

```
vtools_report meta_analysis ceu/EA_RV_VT.asso.res yri/YA_RV_VT.asso.res --beta 5 --pval 6 --
se 7 -n 2 --link 1 > META_RV_VT.asso.res
```

To view the results,

```
cut -f1,3 META_RV_VT.asso.res | head
```

refgene.name2	pvalue meta
CASP7	4.751E-01
POLR2J2	3.110E-01
GNAO1	6.875E-02
C18orf25	9.456E-01
GBP7	3.498E-01
MSH5	5.905E-01
OR51B5	5.521E-01
MAPK14	3.063E-01
BAZ2B	7.941E-01

Note that for genes that only appears in one study but not the other, or only have a valid p-value in one study but not the other, will be ignored from meta-analysis.

2.8 Summary

Analyzing variants with VAT is much like any other analysis software with a general workflow of:

- Variant level cleaning
- Sample genotype cleaning
- Variant annotation and phenotype information processing
- Sample/variant selection
- Association analysis
- Interpreting the findings

The data cleaning and filtering conditions within this exercise should be considered as general guidelines. Your data may allow you to be laxer with certain criteria or force you to be more stringent with others.

Questions

Question 1 List the four lowest p-values and associated variants or gene regions for the EA CV.asso.res, EA RV.asso.res, and EA RV VT.asso.res test outputs, which are results from single variant Wald test, rare variant BRV and VT tests, respectively, using the European American (CEU) population. Also, list the results using Yoruba African (YRI) population from YA CV.asso.res, YA RV.asso.res and YA RV VT.asso.res

EA_CV.asso.res - single variant tests using CEU

1) _____; 2) _____

3) _____; 4) _____

EA_RV.asso.res - BRV tests using CEU

1) _____; 2) _____

3) _____; 4) _____

EA_RV_VT.asso.res - VT tests using CEU

1) _____; 2) _____

3) _____; 4) _____

YA_CV.asso.res - single variant tests using YRI

1) _____; 2) _____

3) _____; 4) _____

YA_RV.asso.res - BRV tests using YRI

1) _____; 2) _____

3) _____; 4) _____

YA_RV_VT.asso.res - VT tests using YRI

1) _____; 2) _____

3) _____; 4) _____

Question 2 List any gene regions that show up in the lowest eight p-values for both the BRV and the VT tests. Why might the p-values for the VT tests be higher than the p-values for the BRV tests? Are any of the top p-value hits significant? Why or why not?

Answers

Question 1

EA CV.asso.res

- 107888886 0.000105185
- 1) 15869257 0.00038548
- 2) 56293401 0.000386273
- 3) 15869388 0.00279873

EA RV.asso.res

- 1) CIDEA 0.00504822
- 2) UGT1A10 0.00549521
- 3) UGT1A5 0.00549521
- 4) UGT1A6 0.00549521

EA_RV_VT.asso.res

- 1) UGT1A9 0.007996
- 2) CPED1 0.00999001
- 3) UGT1A10 0.00999001
- 4) UGT1A6 0.011988

YA CV.asso.res

- 1) 107888886 0.00000974
- 2) 6003506 0.000211457
- 3) 25901623 0.001329
- 4) 3392651 0.00194995

YA RV.asso.res

- 1) EMILIN2 0.00262487
- 2) ASIC2 0.0551664
- 3) MDN1 0.0593085
- 4) BAZ2B 0.0607625

YA_RV_VT.asso.res

- 1) EMILIN2 0.00533156
- 2) MDN1 0.013986
- 3) VLDLR 0.01998
- 4) LRRC9 0.025974

Question 2: The p-values do not achieve significance based on the corrected p values above (Bonferroni correction for multiple tests). Since the BMI values were randomly generated for each individual it is unlikely that any of the p-values for the single variant and aggregation tests would have achieved significance. Also, because of the multiple testing, the p-values for the VT tests might be higher than the p-values for the BRV tests.

References

- [1] Wang, G.T., Peng, B., and Leal, S.M. (2014). Variant Association Tools for Quality Control and Analysis of Large-Scale Sequence and Genotyping Array Data. *Am. J. Hum. Genet.* 94, 770783
- [2] Li B, Leal SM. Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. *Am J Hum Genet* 2008 83:311-21
- [3] Auer PL, Wang G, Leal SM. Testing for rare variant associations in the presence of missing data. *Genet Epidemiol* 2013 37:529-38
- [4] Liu DJ, Leal SM. A novel adaptive method for the analysis of next-generation sequencing data to detect complex trait associations with rare variants due to gene main effects and interactions. *PLoS Genet* 2010 6:e1001156
- [5] Madsen BE, Browning SR. A groupwise association test for rare mutations using a weighted sum statistic. *PLoS Genet* 2009 5:e1000384
- [6] Price AL, Kryukov GV, de Bakker PI, Purcell SM, Staples J, Wei LJ, Sunyaev SR. Pooled association tests for rare variants in exon-resequencing studies. *Am J Hum Genet* 2010 86:832-8
- [7] Wu MC, Lee S, Cai T, Li Y, Boehnke M, Lin X. Rare-variant association testing for sequencing data with the sequence kernel association test. *Am J Hum Genet* 2011 89:82-93
- [8] Lucas FAS, Wang G, Scheet P, Peng B. Integrated annotation and analysis of genetic variants from next-generation sequencing studies with variant tools. *Bioinformatics* 2012 28:421-2
- [9] Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res* 2010 38:e164
- [10] Manichaikul A, Mychaleckyj JC, Rich SS, Daly K, Sale M, Chen WM. Robust relationship inference in genome-wide association studies. *Bioinformatics* 2010 26(22):2867-2873
- [11] Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC. PLINK: a toolset for whole-genome association and population-based linkage analysis. *Am J Hum Genet*, 2007 81:559-75

Association Analysis of Sequence Data using PLINK/SEQ (PSEQ)

Copyright (c) 2022 Stanley Hooker, Biao Li, Di Zhang and Suzanne M. Leal

Purpose

PLINK/SEQ (PSEQ) is an open-source C/C++ library for working with human genetic variation data. The specific focus is to provide a platform for analytic tool development for variation data from large-scale resequencing and genotyping projects, particularly whole-exome and whole-genome studies. PSEQ is independent of, but designed to be complementary to, the existing PLINK (Purcell *et al.*, 2007) package. Here we give an overview of analysis of exome sequence data using PSEQ.

Software Resource

This tutorial was completed with PSEQ 0.10, (released on 14-Jul-2014) available from <https://atgu.mgh.harvard.edu/plinkseq/download.shtml>. Links to PSEQ documentation can also be found on the webpage. Below is an outline of what PSEQ documentation offers:

- Basic Syntax and Conventions
- Project Management
- Data Input
- Attaching Auxiliary Data
- Viewing Data
- Data Output
- Summary Statistics
- Association Analysis
- Locus Database Operations
- Reference Database Operations
- Miscellaneous commands

Exercise Genotype Data

Autosomal exome genotype data was downloaded from the 1000 Genomes pilot data July 2010 release for both the CEU (Utah residents with Northern and Western European ancestry) and YRI (Yoruba in Ibadan, Nigeria) populations. The data sets (CEU.exon.201003.genotypes.vcf.gz and YRI.exon.201003.genotypes.vcf.gz) are available from:

`ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/release/2010_07/exon/snps`

The genomic co-ordinate for this data set is hg18 based. To use the PSEQ annotation data source which is hg19 based, you will lift over this data set to use hg19 co-ordinate. Since PSEQ does not provide a liftover feature therefore the data has already been lifted over for you using Variant Association Tools. The resulting data files, **CEU.exon.201003.genotypes.hg19.vcf.gz** and **YRI.exon.201003.genotypes.hg19.vcf.gz**, will be used for this exercise. One data set contains exome data for European-Americans (CEU) from 1000 Genomes while the other for Yoruba (YRI). The liftover feature may also have to be used with your data set as new hg coordinates become available. For additional information see <http://varianttools.sourceforge.net/Vtools/Liftover>

Phenotype Data

To demonstrate performing an association analysis, we simulated a quantitative trait phenotype (BMI). Please note that these phenotypes are **NOT** from the 1000 genomes project. The phenotype data for the

exercise can be found in the text file **phenotype.phe**. This phenotype file contains data for 202 individuals from both the CEU and YRI populations.

Computation Resources

The following tutorial uses a small data set so that the association analysis can be completed in a short period-of-time. Large next-generation sequenced data sets require a reasonably powerful machine with a high-speed internet connection.

Data Cleaning and Variant/Sample Selection

Getting Started

To get a list of PSEQ subcommands use:

```
pseq help
```

Or,

```
pseq help all
```

Create a new project

```
pseq myproj new-project --resources hg19
```

Creating new project specification file [myproj.pseq]

The “--resources” flag tells **pseq** where your supporting databases are located. For this exercise the necessary databases have already been created and are within your exercise directory. Instructions on how to create these databases is located at:

<http://atgu.mgh.harvard.edu/plinkseq/resources.shtml>.

Load variant data

Import all vcf files under the current directory:

```
pseq myproj load-vcf --vcf CEU.exon.2010_03.genotypes.hg19.vcf.gz YRI.exon.2010_03.genotypes.hg19.vcf.gz
loading : /home/gmc01/data/pseq/CEU.exon.2010_03.genotypes.hg19.vcf.gz ( 90 individuals )
parsed 3000 rows
loading : /home/gmc01/data/pseq/YRI.exon.2010_03.genotypes.hg19.vcf.gz ( 112 individuals )
parsed 5000 rows
/home/gmc01/data/pseq/CEU.exon.2010_03.genotypes.hg19.vcf.gz : inserted 3489 variants
/home/gmc01/data/pseq/YRI.exon.2010_03.genotypes.hg19.vcf.gz : inserted 5175 variants
```

Note CEU are European-Americans and YRI are Yoruba from Nigeria.

Load phenotype data

```
pseq myproj load-pheno --file phenotype.phe
```

Processed 202 rows

The “phenotype.phe” file contains phenotypes for SEX, BMI and RACE (BMI is body mass index, males are denoted by a 1 and females by 2). Instruction on formatting .phe file can be found at <https://atgu.mgh.harvard.edu/plinkseq/input.shtml#phe>.

View variants and samples

To view variant sites info:

```
pseq myproj v-view | head
```

chr1:1115461	.	C/T	.	1	PASS
chr1:1115503	.	T/C	.	1	SBFilter
chr1:1115510	.	C/T	.	1	PASS
chr1:1115548	.	G/A	.	1	PASS
chr1:1115604	.	C/A	.	1	PASS
chr1:1118275	rs61733845	C/T	.	2	PASS
chr1:1119399	.	C/T	.	1	PASS
chr1:1119434	.	C/A	.	1	PASS
chr1:1120370	.	C/G	.	1	PASS
chr1:1120377	.	T/A	.	1	PASS

v-view command outputs a per-variant level view of a project, with the above fields: chromosome (base-position); variant-ID (or '.' If novel); ref/alt alleles; a sample/file identifier (or '.' If consensus variant); # of samples the variant observed in; filter values for samples (here 'PASS' means that the variant site passes all filter and 'SBFilter' means that the variant site fails to pass the strand bias (SB) filter). More details about v-view command can be found at <https://atgu.mgh.harvard.edu/plinkseq/view.shtml#var>

To view samples and phenotypes:

i-view command writes to standard output to view individuals' phenotype information

```
pseq myproj i-view | head
```

```
#BMI (Float) "BMI"
#RACE (String) "RACE"
#SEX (Integer) "SEX"
#PHE .
#STRATA .
#ID FID IID MISS SEX PAT MAT META
NA06984 . . 0 0 . . BMI=36.353;RACE=CEU;SEX=1
NA06985 . . 0 0 . . BMI=21.415;RACE=CEU;SEX=2
NA06986 . . 0 0 . . BMI=26.898;RACE=CEU;SEX=1
NA06989 . . 0 0 . . BMI=25.015;RACE=CEU;SEX=2
```

There are 3 fields, BMI, RACE and SEX contained in the input phenotype file, phenotype.phe. The headers are #ID – main unique individual ID; FID – optional family ID; IID: optional individual ID; MISS – a flag to indicate missing data; SEX – sex; PAT – paternal ID; MAT – maternal ID; META – meta information of fields from input phenotype file. More details about i-view command outputs can be found at <https://atgu.mgh.harvard.edu/plinkseq/view.shtml#ind>.

Summary

To view a summary of the complete project

```
pseq myproj summary
```

Command above will generate a long list of output. To view summaries of portions of the project, i.e., variant data, phenotype data, locus data, reference data, sequence data, input files and meta data:

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
```

```
6987 unique variants
File tag : 1 (3489 variants, 90 individuals)
File tag : 2 (5175 variants, 112 individuals)
```

pseq myproj ind-summary

-----Individual DB summary-----

202 unique individuals
Phenotype : BMI (Float) "BMI"
Phenotype : RACE (String) "RACE"
Phenotype : SEX (Integer) "SEX"

pseq myproj loc-summary

pseq myproj ref-summary

pseq myproj seq-summary

pseq myproj file-summary

pseq myproj meta-summary

More details about viewing summary information for project databases can be found at <https://atgu.mgh.harvard.edu/plinkseq/proj.shtml#summ>

Based on the “pseq myproj var-summary” command there are 6987 unique variant sites for CEU and YRI, with the CEU sample having 3489 variant sites and the YRI sample 5175 variant sites. .

For an overview of variant summary statistics:

pseq myproj v-stats

NVAR 6987
RATE 0.568384
MAC 19.8557
MAF 0.0691347
SING 2064
MONO 30
TITV 3.57264
TITV_S 3.77778
DP 8426.74
QUAL NA
PASS 0.999857
FILTER|PASS 0.999857
FILTER|SBFilter 0.000143123
PASS_S 1

v-stats command obtains summary statistics across variants. Output statistics are NVAR – total number of variants; RATE – average call rate; MAC – mean minor allele count; MAF – mean minor allele frequency; SING – number of singletons; MONO – number of monomorphic sites; TITV – transition/transversion (Ti/Tv) ratio; TITV_S – Ti/Tv ratio for singletons; DP – mean variant read depth; QUAL – mean QUAL score from VCF; PASS – proportion of variants that PASS all FILTERS; FILTER|PASS – proportion of variants that pass all filters; FILTER|SBFilter – proportion of variants that fail to pass SB filter. More details about v-stats command outputs can be found at <https://atgu.mgh.harvard.edu/plinkseq/stats.shtml#var>

For individual level summary statistics:

pseq myproj i-stats | head

ID	NALT	NMIN	NHET	NVAR	RATE	SING	TITV	PASS	PASS_S	QUAL	DP
NA06984	719	568	480	3162	0.452555	8	3.61789	568	8	NA	13489
NA06985	655	531	420	3144	0.449979	10	3.5	531	10	NA	13530.3
NA06986	773	643	503	3437	0.491914	22	3.69343	643	22	NA	12535.8
NA06989	699	532	469	3130	0.447975	8	3.22222	532	8	NA	13549.7
NA06994	591	464	377	3002	0.429655	3	3.59406	464	3	NA	13923.8
NA07000	802	613	517	3388	0.484901	10	3.67939	613	10	NA	12292.6
NA07037	800	631	512	3374	0.482897	4	3.60584	631	4	NA	12357.4
NA07048	817	675	607	3373	0.482754	15	3.29936	675	15	NA	12909.5
NA07051	825	637	507	3451	0.493917	13	3.05732	637	13	NA	11929

i-stats command obtains a matrix of summary statistics for every individual in a project. Output statistics are ID – individual ID; NALT – number of non-reference genotypes; NMIN – number of genotypes with a minor allele; NHET – number of heterozygous genotypes for individual; NVAR – total number of called

variants for individual; RATE – genotyping rate for individual; SING – number of singletons individuals has; TITV – mean Ti/Tv for variants for which individual has a nonreference genotype; PASS – number of variants passing for which individual has a nonreference genotype; PASS_S - number of singletons passing for which individual has a (singleton) nonreference genotype; QUAL - mean QUAL for variants for which individual has a nonreference genotype; DP - mean variant DP for variants for which individual has a nonreference genotype. More details about i-stats command output can be found at <https://atgu.mgh.harvard.edu/plinkseq/stats.shtml#ind>

The file tags (listed at the top of the “pseq myproj var-summary” results as “1” for the CEU imported VCF file and “2” for YRI imported VCF file) can be changed to more identifiable names using the commands:

```
pseq myproj tag-file --id 1 --name CEU
```

```
pseq myproj tag-file --id 2 --name YRI
```

To view changes use the command:

```
pseq myproj var-summary
-----Variant DB summary-----
```

```
6987 unique variants
File tag : CEU (3489 variants, 90 individuals)
File tag : YRI (5175 variants, 112 individuals)
```

This will help us later for viewing population specific data as well as filtering and analyzing data based on population.

Variant statistics

Variant statistics such as Hardy-Weinberg equilibrium, minor allele count, and minor allele frequency can be output using the “v-freq” command:

```
pseq myproj v-freq | head
```

VAR	CHR	POS	REF	ALT	FILTER	QUAL	TI	GENO	MAC	MAF	REFMIN	HWE	HET	NSNP
chr1:1115461	1	1115461	C	T	PASS	.	1	0.311881	4	0.031746	0	1	0.0634921	3
chr1:1115503	1	1115503	T	C	SBFilter	.	1	0.282178	4	0.0350877	0	1	0.0701754	2
chr1:1115510	1	1115510	C	T	PASS	.	1	0.331683	2	0.0149254	0	1	0.0298507	2
chr1:1115548	1	1115548	G	A	PASS	.	1	0.262376	1	0.00943396	0	1	0.0188679	1
chr1:1115604	1	1115604	C	A	PASS	.	0	0.287129	3	0.0258621	0	1	0.0517241	0
chr1:1118275	1	1118275	C	T	PASS	.	1	0.579208	45	0.192308	0	0.367544	0.282051	0
chr1:1119399	1	1119399	C	T	PASS	.	1	0.49505	3	0.015	0	1	0.03	1
chr1:1119434	1	1119434	C	A	PASS	.	0	0.49505	1	0.005	0	1	0.01	0
chr1:1120370	1	1120370	C	G	PASS	.	0	0.49505	16	0.08	0	0.478564	0.14	2

Please note that it is not valid to filter for deviation from HWE using the entire project since there are two populations, instead the HWE much be examined for each individual project.

For population specific variant statistics use the “--mask” flag with the “file” option:

```
pseq myproj v-freq --mask file=CEU | head
```

VAR	CHR	POS	REF	ALT	FILTER	QUAL	TI	GENO	MAC	MAF	REFMIN	HWE	HET	NSNP
chr1:1115503	1	1115503	T	C	SBFilter	0	1	0.633333	4	0.0350877	0	1	0.0701754	1
chr1:1115548	1	1115548	G	A	PASS	0	1	0.588889	1	0.00943396	0	1	0.0188679	0
chr1:1118275	1	1118275	C	T	PASS	0	1	0.677778	3	0.0245902	0	1	0.0491803	0
chr1:1120377	1	1120377	T	A	PASS	0	0	0.988889	1	0.00561798	0	1	0.011236	1
chr1:1120431	1	1120431	G	A	PASS	0	1	0.855556	6	0.038961	0	1	0.0779221	0
chr1:3548136	1	3548136	T	C	PASS	0	1	0.811111	18	0.123288	1	1	0.219178	0
chr1:3548832	1	3548832	G	C	PASS	0	0	0.988889	13	0.0730337	0	1	0.146067	0
chr1:3551737	1	3551737	C	T	PASS	0	1	0.988889	1	0.00561798	0	1	0.011236	1
chr1:3551792	1	3551792	G	A	PASS	0	1	1	8	0.0444444	0	1	0.0888889	0

```
pseq myproj v-freq --mask file=YRI | head
```

VAR	CHR	POS	REF	ALT	FILTER	QUAL	TI	GENO	MAC	MAF	REFMIN	HWE	HET	NSNP
chr1:1115461	1	1115461	C	T	PASS	0	1	0.5625	4	0.031746	0	1	0.0634921	1
chr1:1115510	1	1115510	C	T	PASS	0	1	0.598214	2	0.0149254	0	1	0.0298507	1
chr1:1115604	1	1115604	C	A	PASS	0	0	0.517857	3	0.0258621	0	1	0.0517241	0
chr1:1118275	1	1118275	C	T	PASS	0	1	0.5	42	0.375	0	0.395585	0.535714	0
chr1:1119399	1	1119399	C	T	PASS	0	1	0.892857	3	0.015	0	1	0.03	1
chr1:1119434	1	1119434	C	A	PASS	0	0	0.892857	1	0.005	0	1	0.01	0
chr1:1120370	1	1120370	C	G	PASS	0	0	0.892857	16	0.08	0	0.478564	0.14	1
chr1:1120431	1	1120431	G	A	PASS	0	1	0.741071	67	0.403614	0	0.360868	0.542169	4
chr1:1120488	1	1120488	A	C	PASS	0	0	0.857143	10	0.0520833	0	1	0.104167	3

As you see, the “--mask” flag is used to set conditions for the viewing or filtering variants or individuals. More details about “v-freq” command can be found at

<https://atgu.mgh.harvard.edu/plinkseq/tutorial.shtml>

Data Cleaning

Removal of low quality variants

To view the number of variants that passed all quality filters:

```
pseq myproj v-view --mask any.filter.ex | head
```

chr1:1115461	.	C/T	.	1	PASS
chr1:1115510	.	C/T	.	1	PASS
chr1:1115548	.	G/A	.	1	PASS
chr1:1115604	.	C/A	.	1	PASS
chr1:1118275	rs61733845	C/T	.	2	PASS
chr1:1119399	.	C/T	.	1	PASS
chr1:1119434	.	C/A	.	1	PASS
chr1:1120370	.	C/G	.	1	PASS
chr1:1120377	.	T/A	.	1	PASS
chr1:1120431	rs1320571	G/A	.	2	PASS

```
pseq myproj v-view --mask any.filter.ex | wc -l
```

There are 6986 unique variant sites that have passed the quality filters. The “--mask” flag gives the condition(s) that must be met for the variant to be listed. Here “any.filter.ex” tells **pseq** to remove any variants that failed 1 or more quality filters. Only variants that have a ‘PASS’ value in the FILTER field of the vcf file will be selected. More details about filtering variants on FILTER field can be found at <https://atgu.mgh.harvard.edu/plinkseq/masks.shtml#filter>

To view the number of variants that failed any quality filter:

```
pseq myproj v-view --mask any.filter | wc -l
```

One variant failed the filter. To select only variants that passed all quality filters:

```
pseq myproj var-set --group pass --mask any.filter.ex
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
```

```
6987 unique variants
File tag : CEU (3489 variants, 90 individuals)
File tag : YRI (5175 variants, 112 individuals)
```

```
Set pass containing 8663 variants
```

The “var-set” option tells **pseq** that we will be creating a new set of variants, the input following the “--group” flag gives the name of the new variant set, and the input following the “--mask” flag gives the condition(s) that must be met for the variant to be included in the new variant set.

If we consider variant sites with a read depth < 15 as low quality variant sites and we want to remove variants that did not meet this threshold. Note that ‘DP’, which denotes total read depth of a variant site, is contained in the INFO field of vcf file.

```
pseq myproj var-set --group pass_DP15 --mask include="DP>14" var=pass
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
```

```
6987 unique variants
File tag : CEU (3489 variants, 90 individuals)
File tag : YRI (5175 variants, 112 individuals)
```

```
Set pass containing 8663 variants
Set pass_DP15 containing 8662 variants
```

Only one variant site is removed. The “var=allpass” option allows us to use a previously defined variant set as a reference for additional filtering of a previously filtered variant set. By using various “--mask” commands you can filter out variants that are not useful for your particular study.

Filter data by genotype read depth 10

```
pseq myproj var-set --group pass_DP15_DPgeno10 --mask geno=DP:ge:11 var=pass_DP15
```

```
pseq myproj var-summary
-----Variant DB summary-----
```

```
6987 unique variants
File tag : CEU (3489 variants, 90 individuals)
File tag : YRI (5175 variants, 112 individuals)
```

```
Set pass containing 8663 variants
Set pass_DP15 containing 8662 variants
Set pass_DP15_DPgeno10 containing 8662 variants
```

This command sets all genotypes with a sequencing depth (DP) < 11 to null using the option “geno=DP:ge:11”. In the vcf file, genotype level DP information is contained in the genotype columns, present under each individual ID and is specific to every individual’s genotype. Available genotype level information is denoted by FORMAT column in the vcf file.

Association Tests for a Quantitative Trait

NOTE: From this step forward the association tests will be performed for the CEU population only. The “file=YRI” tag can be used to perform the same tests on the YRI data.

Select CEU variant sites

```
pseq myproj var-set --group pass_DP15_DPgeno10_CEU --mask file=CEU var=pass_DP15_DPgeno10
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
...
Set pass_DP15_DPgeno10_CEU containing 3488 variants
```

There are 3488 variant sites that can be found in CEU population dataset after QC.

Exclude variant sites with HWE p-value < 5.7e-7

```
pseq myproj var-set --group pass_DP15_DPgeno10_CEU_HWE --mask hwe=5.7e-7:1 var=pass_DP15_DPgeno10_CEU
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
...
Set pass_DP15_DPgeno10_CEU containing 3479 variants
```

There are 3479 variant sites that are in HWE (Hardy-Weinberg equilibrium) in CEU population. Details about tests for deviation from HWE can be found at http://en.wikipedia.org/wiki/Hardy-Weinberg_principle. Here we use a p-value cutoff of 5.7e-7 to exclude variant sites, for more details see reference <http://www.nature.com/nature/journal/v447/n7145/full/nature05911.html>

Filter variants by minor allele frequency (MAF)

We wish to analyze variant sites with different allele frequencies. In order to obtain the different data sets the following commands are used.

To extract variant sites with $MAF \geq 0.05$:

```
pseq myproj var-set --group pass_DP15_DPgeno10_CEU_HWE_MAFgt05 --mask maf=0.05:0.5
var=pass_DP15_DPgeno10_CEU_HWE
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
...
Set pass_DP15_DPgeno10_CEU_HWE_MAFgt05 containing 1429 variants
```

There are 1429 variant sites in the CEU data set that pass QC with a $MAF \geq 0.05$. These variant sites are saved to the variant table; pass_DP15_DPgeno10_CEU_HWE_MAFgt05.

To extract variant sites with $MAF \leq 0.01$:

```
pseq myproj var-set --group pass_DP15_DPgeno10_CEU_HWE_MAFlt01 --mask "mac=1 maf=0.01"
var=pass_DP15_DPgeno10_CEU_HWE
```

```
pseq myproj var-summary
```

```
-----Variant DB summary-----
Set pass_DP15_DPgeno10_CEU_HWE_MAFlt01 containing 1083 variants
```

There are 1083 variant sites in the CEU dataset which pass QC with a $MAF \leq 0.01$. The variant sites are saved to the variant table; pass_DP15_DPgeno10_CEU_HWE_MAFlt01. Note that condition “mac=1” excludes monomorphic sites.

More details about --mask options on filtering variants on sample polymorphism can be found at <https://atgu.mgh.harvard.edu/plinkseq/masks.shtml#maf>

Analysis of common variants ($MAF \geq 0.05$)

To run a linear or logistic regression on each single variant, use the glm command. The type of test will depend on the phenotype (quantitative trait or dichotomous disease trait).

To detect single variant association between quantitative phenotype BMI, controlling for sex and a group of variants, contained in variant table pass_DP15_DPgeno10_CEU_HWE_MAFgt05, filtered using each of the previous filtering conditions:

```
pseq myproj glm --phenotype BMI --covar SEX --mask var=pass_DP15_DPgeno10_CEU_HWE_MAFgt05 > SNV_CEU.result
```

```
head SNV_CEU.result
```

VAR	REF	ALT	N	F	BETA	SE	STAT	P
chr1:3548136	T	C	73	0.876712	-1.5394	1.85033	-0.83087	0.40998
chr1:3548832	G	C	89	0.0730337	1.13049	2.26738	0.49859	0.619341
chr1:6524501	T	C	86	0.0697674	0.433904	2.49357	0.174009	0.862282
chr1:6524688	T	C	88	0.0511364	-1.8695	2.70494	-0.68658	0.491718
chr1:11710561	T	G	47	0.117021	-0.3445	1.92692	-0.38637	0.857716
chr1:17914057	G	A	86	0.0755814	-1.5906	2.34734	-0.67942	0.498754
chr1:17914122	G	A	85	0.0823529	2.61561	2.1748	1.20269	0.232558
chr1:17961345	C	T	68	0.110294	2.99054	2.00047	1.49492	0.139775
chr1:17981184	A	C	80	0.15	-1.8308	1.63531	-1.11972	0.266315

The output statistics are VAR – variant identifier; REF – reference allele; ALT – alternate allele(s); N – number of individuals included in analysis; F – frequency of the alternate allele(s); BETA – regression coefficient; SE – standard error of estimate; STAT – test statistic; P – asymptotic p-value. More details about linear and logistic regression models can be found at <https://atgu.mgh.harvard.edu/plinkseq/assoc.shtml#glm>

To view the results sorted by p-value:

```
cat SNV_CEU.result | awk '{if(FNR==1) print $0; if(NR>1) print $0 | "sort -k9"}' | grep -v "NA\s\++NA\s\++NA" | head
```

VAR	REF	ALT	N	F	BETA	SE	STAT	P
chr11:108383676	A	G	90	0.138889	6.36308	1.60942	3.95365	0.000156342
chr19:16008388	A	C	53	0.122642	6.88317	1.73915	3.95778	0.000239339
chr19:16006413	G	A	80	0.1	6.31788	1.78167	3.54604	0.000669193
chr14:39901157	C	A	36	0.0555556	10.8531	3.12283	3.47542	0.00144933
chr16:57735900	G	C	80	0.29375	-4.1014	1.43663	-2.9009	0.004718
chr2:49189921	C	T	90	0.588889	-3.36	1.17772	-2.8405	0.0056123
chr7:156742501	C	G	9	0.277778	-12.1591	2.89402	-4.2040	0.00567644
chr2:49191041	C	T	89	0.58427	-3.3654	1.19515	-2.0240	0.00607226
chr15:25926204	C	G	83	0.0783133	5.79532	2.13611	2.71302	0.00816109

Analysis of rare variants (MAF ≤ 0.01)

PSEQ has a collection of gene-based tests, see <https://atgu.mgh.harvard.edu/plinkseq/assoc.shtml#genic> for details.

However, Currently only the SKAT and SKAT-O can be used to analyze quantitative traits so the SKAT test will be used in the following rare variant burden analysis (if we choose to use other tests, e.g. WSS – frequency-weighted test, VT – variable threshold test, etc., the following error will be returned.

```
pseq myproj assoc --tests fw vt --phenotype BMI
pseq error : only SKAT/SKAT--O can handle quantitative traits
```

To perform SKAT, where rare variants aggregated across a gene region, a group-by mask is required. Here we use `loc.group=refseq`, where `refseq` denotes NCBI Reference Sequence Database. More details about grouping variants can be found at <https://atgu.mgh.harvard.edu/plinkseq/masks.shtml#groups>. More details about `refseq` can be found at <http://www.ncbi.nlm.nih.gov/refseq/>

When performing single variant analysis data QC can be performed and then variant table containing selected variants can be analyzed. If a rare variant aggregate association test is being performed it is not possible using PSEQ to specify the name of the variant table, instead all of the QC parameters must be included in the command line in addition to the association test parameters.

Running the SKAT test using the variant table results in an error:

```
pseq myproj assoc --tests skat --phenotype BMI --covar SEX --mask var=pass_DP15_DPgeno10_CEU_HWE_MAF101
loc.group=refseq > SKAT_CEU.result
```

```
pseq error : you cannot specify other includes in the mask with loc.group
```

Additional details can be found at <https://atgu.mgh.harvard.edu/plinkseq/whatisnew.shtml>),

Although we use the most recent version `pseq-0.10` in this exercise (for which there is no updated documentation), the error still remains unresolved. Therefore, we have to redo cleaning on original data by re-specifying each filtering condition and run SKAT using one command as below:

```
pseq myproj assoc --tests skat --phenotype BMI --covar SEX --mask include="DP>14" geno=DP:ge:11 file=CEU hwe=5.7e-7:1
"mac=1 maf=0.01" loc.group=refseq > SKAT_CEU.result
```

head -20 SKAT_CEU.result

LOCUS	POS	ALIAS	NVAR	TEST	P	I	DESC
NM_000055	chr3:165548187	G/A	W=1	0:0			
NM_000055	chr3:165548187..165548187	BCHE	1	SKAT	0.237374	.	.
NM_000112	chr5:149359938	C/G	W=1	0:0			
NM_000112	chr5:149360143	T/C	W=1	0:0			
NM_000112	chr5:149360212	A/G	W=1	0:0			
NM_000112	chr5:149360215	T/C	W=1	0:0			
NM_000112	chr5:149361245	G/A	W=1	0:0			
NM_000112	chr5:149359938..149361245	SLC26A2	5	SKAT	0.293096	.	.
NM_000119	chr15:43498537	C/T	W=1	0:0			
NM_000119	chr15:43499436	G/A	W=1	0:0			
NM_000119	chr15:43500478	C/T	W=1	0:0			
NM_000119	chr15:43498537..43500478	EPB42	3	SKAT	0.422114	.	.
NM_000122	chr2:128016983	C/T	W=1	0:0			
NM_000122	chr2:128038204	T/C	W=1	0:0			

NM_000122	chr2:128016983..128038204	ERCC3	2	SKAT	0.386466	.	.
NM_000124	chr10:50732644	G/C	W=1	0:0			
NM_000124	chr10:50738781	T/C	W=1	0:0			
NM_000124	chr10:50740844	G/A	W=1	0:0			
NM_000124	chr10:50740861	C/T	W=1	0:0			

For each gene region the list of the variants within the gene are listed, followed by gene-based association results. The I field is only available for case control data and provides the smallest possible empirical p-value which can be obtained for the variant sites and the DESC field which is also only available for case control data and it provides the number of case and control alternative alleles. Since we are analyzing quantitative trait data these fields are blank. Detailed explanation about each output field can be found at <https://atgu.mgh.harvard.edu/plinkseq/assoc.shtml#genic>

To view the smallest p-values for each SKAT test:

```
cat SKAT_CEU.result | grep SKAT | grep -v "P=NA" | sort -k6 | head -15
```

NM_024837	chr15:50152449..50264848	ATP8B4	5	SKAT	0.00405073	.	.
NM_001055	chr16:28617413..28617413	SULT1A1	1	SKAT	0.00418122	.	.
NM_177529	chr16:28617413..28617413	.	1	SKAT	0.00418122	.	.
NM_177530	chr16:28617413..28617413	.	1	SKAT	0.00418122	.	.
NM_177534	chr16:28617413..28617413	.	1	SKAT	0.00418122	.	.
NM_177536	chr16:28617413..28617413	.	1	SKAT	0.00418122	.	.
NM_001137559	chr12:121746337..121764935	ANAPC5	3	SKAT	0.00621198	.	.
NM_016237	chr12:121746337..121764935	.	3	SKAT	0.00621198	.	.
NM_006371	chr3:33174163..33174163	CRTAP	1	SKAT	0.00748816	.	.
NM_006944	chr2:234959642..234967570	SPP2	3	SKAT	0.00753125	.	.
NM_018328	chr2:149221327..149241000	MBD5	4	SKAT	0.00755692	.	.
NM_000782	chr20:52779338..52779338	CYP24A1	1	SKAT	0.00794735	.	.
NM_001128915	chr20:52779338..52779338	.	1	SKAT	0.00794735	.	.
NM_001018088	chr15:62204043..62302757	.	3	SKAT	0.0221564	.	.
NM_017684	chr15:62204043..62302757	VPS13C	3	SKAT	0.0221564	.	.

Note that each test has been performed on each alternative transcript (NM_*) of each gene, e.g. transcripts NM_001055, NM_177529, NM_177530, NM_177534 and NM_177536 all belong to gene SULT1A1.

Questions

Repeat the above analysis but using the data from the Yoruba (YRI) population and answer the following questions.

Question 1

List the four smallest p-values for the single variant tests for the common variants i.e. $MAF \geq 0.05$:

- 1.) _____
- 2.) _____
- 3.) _____
- 4.) _____

List the four smallest p-values for the SKAT rare variant test:

- 1.) _____
- 2.) _____
- 3.) _____
- 4.) _____

Answers

Question 1

Single variant test

- 1.) ____ chr21:26979752____ 0.00084882____
- 2.) ____ chr17:3445901 ____ 0.000956475____
- 3.) ____ chr17:9729445____ 0.0010022____
- 4.) ____ chr19:15303225____ 0.0011692____

SKAT aggregate burden test

- 1.) ____ NM_207317____ 0.0210752____
- 2.) ____ NM_032048____ 0.0238947____
- 3.) ____ NM_002738____ 0.0255961____
- 4.) ____ NM_212535____ 0.0255961____

Sample Size Calculations - Cochran-Armitage Test for Trend

Copyrighted © 2022 Suzanne M. Leal

Webpage for the exercises:

http://csg.sph.umich.edu/abecasis/cats/gas_power_calculator/index.html

<http://ihg.helmholtz-muenchen.de/cgi-bin/hw/power2.pl>

<http://zzz.bwh.harvard.edu/gpc/cc2.html>

Question 1

For a complex disease study, you plan to collect 35,000 cases and 70,000 controls and wish to know if this is a sufficient sample size to detect associations with disease susceptibility loci. The disease has a population prevalence of 5%. You wish to estimate the power for a genotypic relative risk of 1.2 and a disease allele frequency of 0.02. What is the power for $\alpha=5 \times 10^{-8}$ under a multiplicative model ($\gamma_2 = \gamma_1^2$) a.) _____ and dominant model ($\gamma_2 = \gamma_1$) b.) _____?

Question 2

For your study, you hypothesize that you will try to replicate associations for 100 variants that are in linkage equilibrium and you want to reject the null hypothesis using a p-value of 0.05. What is the Bonferroni correction you should use a.) _____. Determine what your power would be if you used a Bonferroni correction to control for the Family Wise Error Rate (FWER) for testing 100 variants. Using the parameters provided in question 1 but for a sample size of 20,000 cases and 20,000 controls what is the power under the multiplicative model b.) _____ and under a dominant model c.) _____?

Question 3

You determine that you can ascertain 50,000 cases and 50,000 controls what is the power using the same parameters as described in question 1 for the multiplicative model _____ and dominant model _____?

Question 4

The power of the Cochran-Armitage test for trend is dependent on the underlying genetic model. Using the parameters from question 1 which of the following underlying genetic models: multiplicative ($\gamma_2 = \gamma_1^2$), additive ($\gamma_2 = 2\gamma_1 - 1$), dominant ($\gamma_2 = \gamma_1$) or recessive ($\gamma_1 = 1$) would you predict to be the most powerful a.) _____ and least powerful b.) _____?

Question 5

For study design with equal numbers of cases and controls a genotype relative risk of 1.5 under a recessive model for a disease with a population prevalence of 0.05 and disease allele frequency of 0.1. How many cases a.) _____ and controls b.) _____ should you ascertain for $\alpha=5.0 \times 10^{-8}$ and $1-\beta=0.80$? *Use power2 or Genetic Power Calculator, GAS power cannot calculate for more than 100,000 cases.

Question 6

You are performing a rare variant association study and you assume that that cumulative frequency of the causal variants in your gene region is 0.01 with every variant having an effect size of 1.4. The disease you are studying has a prevalence of 5%. For a study with 0.8 power and an $\alpha=2.5 \times 10^{-6}$ under a dominant model for equal numbers of cases and controls what is the total sample size a.) _____ do you need to ascertain. What is the total sample size b.) _____ you need to ascertain if the cumulative frequency of causal variants is only 0.005?

Question 7

You are performing a study using the UK Biobank and for your phenotype of interest you have 50,000 cases and 100,000 controls. For a disease with 10% prevalence, disease allele frequency of 0.01, where each variant has an effect size of 1.2 under a dominant model what would be the power for an aggregate test where the cumulative allele frequency is 0.01 _____ and a single variant test _____? Clue use the appropriate alpha for each test.

Question 8

Using have a replication sample of 50,000 cases and 50,000 controls and you plan to try to replicate 15 genes and 100 variants. Using the same parameters as in question 7 what would be your power to replicate a.) _____? Note for alpha use a Bonferroni correction.

Question 9

For the above power calculations, you have been using the relative risk which only approximates the odds ratio when a.) _____. You are performing a power calculation for a case control study for a disease/variant frequency of 0.01. You use a dominant model and a gamma of 1.2 for a disease with a prevalence for 0.2. What is the odds ratio for which the power calculations are being performed b.) _____. *Use Genetic Power Calculator – information not provided by GAS or Power2.

ANSWERS

1. a.) 0.702 b.) 0.654
2. a.) 5.0×10^{-4} b.) 0.690 c.) 0.657
3. a.) 0.798 b.) 0.755
4. a.) multiplicative b.) recessive
5. a.) 170,910 b.) 170,910
6. a.) ~43,000 b.) ~84,300
7. a.) 0.73 b.) 0.45 Hint: use $\alpha = 5 \times 10^{-8}$ for single variant test and $\alpha = 2.5 \times 10^{-6}$ for the aggregate test
8. a.) 0.87 (Hint: use $\alpha = 4.3 \times 10^{-4}$)
9. a.) only for disease with low prevalence does the relative risk does not estimate the odds ratio b.) 1.26

Exercise

Multiple Testing

Simultaneous testing of several hypotheses increases the probability to observe at least one significant result by chance. The single-test significance levels (or, correspondingly, the P -values) have to be corrected for this multiplicity. Correction can either aim at controlling the *number* of false-positives (i.e. the family-wise error rate; FWER) or the *proportion* of false-positives (i.e. the false discovery rate; FDR). Corrections can be made following either a stepwise procedure (single-step, step-down, etc.) or by permutation.

In this exercise, stepwise as well as permutation-based correction will be applied. All methods in this exercise do not adjust the single-test significance level, but instead *adjust the P -value* by multiplying it with some correction factor. Thus, all adjusted P -values smaller than the pre-set experiment-wise significant level (usually 0.05) can be considered significant after correction for multiple testing. Please also answer the questions at the end.

Attention: PLINK expects each command to be in a single line! PLINK ignores arguments on subsequent lines after a line break. Please type each command without a line break or use a backslash ('\') before a line break. A backslash causes PLINK to ignore the line break.

Correction with PLINK

Please change the working directory as requested. You are provided with a data set on diastolic blood pressure and the genotypes of 20 SNP markers. The data are already in binary PLINK format. There are three files

- **dbp.fam:** Pedigree file with information on family, sex and affection status
- **dbp.bim:** SNP marker description
- **dbp.bed:** SNP genotypes (in compressed, binary form)

Stepwise correction

Run an association analysis where you test each marker in the data file for association with the case-control status under an allelic genetic risk model. To this end, use the `--assoc` flag of PLINK. Since you perform multiple tests, you will also request stepwise multiple testing corrections by additionally using the `--adjust` flag. Write the results to files named `multttest.*`:

```
plink --bfile dbp --assoc --adjust --out multttest
```

Use a text editor (e.g. notepad/Wordpad under Windows, pico/vi/nano/emacs under Linux) to evaluate the contents the results file 'multttest.assoc.adjusted'. The markers in this file are sorted in ascending order by their "raw", unadjusted P -values. The further columns list the adjusted P -values for a number of correction methods.

Correction by permutation

Now run two separate permutation corrections (the 'Westfall & Young' approach), one with 5000 and one with 100,000 permutations:

```
plink --bfile dbp --assoc --mperm 5000 --out multperm5000
plink --bfile dbp --assoc --mperm 100000 --out multperm100000
```

Inspect the resulting files, `multperm5000.assoc.mperm` and `multperm100000.assoc.mperm`, with a text editor and answer the questions below.

Correction with R

In this exercise, only stepwise correction will be applied. Please also answer the questions at the end.

Start R and change the working directory as requested. P -values for the diastolic blood pressure have already been calculated and stored in the R archive file 'p.values.R'. Load the P -values for the exercise into the R working memory:

```
load("p.values.R")
ls()
p.values
```

Loading a dedicated R library

The function that calculates multiplicity-corrected P -values is contained in a *library* (or *package*). This is a bundle of functions and/or data sets for use in R. Libraries have to be loaded into the R working memory, before their functions can be used. The library `multtest` contains various functions for multiple-testing correction and is available from Bioconductor.

Load the library into working memory with the following command:

```
library(multtest)
```

If this command results in an error message that the library is not installed:

You have to install the `multtest` library on your computer. Under Linux, please ask your system administrator. Under Windows (given you have Administrator privileges), please follow the following steps:

1. Install Bioconductor core packages on your computer by typing the following commands in the R shell:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install()
```

2. Install the `multtest` package:

```
BiocManager::install("multtest")
```

3. Load the `multtest` package:

```
library(multtest)
```

Stepwise correction

The function for multiplicity correction is called `mt.rawp2adjp`. It expects a list of “raw” P -values as well as a vector of the names of those methods that should be applied.

Call this function and assign the result to an object `adj.p.values`. Then print this object:

```
adj.p.values = mt.rawp2adjp(p.values, c("Bonferroni", "Holm", "SidakSS", "BH"))
adj.p.values
```

The object is an R list that contains two elements: a matrix of P -values (`adjp`) and an index (`index`). The markers have been sorted by their “raw” P -values and this order is given in the index. For convenience, we will name the rows of the P -value matrix by the corresponding marker name and then print the object again. The column `EMP2` contains the adjusted empirical P -values:

```
rownames(adj.p.values$adjp) = names(p.values[adj.p.values$index])
adj.p.values$adjp
```

Questions

1. Please enter the raw and stepwise adjusted P -values for markers rs1112 and rs1117 from the analysis in PLINK in the table below.

Marker	Raw P-value UNADJ	Adjusted P -value following			
		BONF [±]	HOLM [¶]	SIDAK_SS [*]	FDR_BH ^{**}
rs1112					
rs1117					

[±] Bonferroni correction

[¶] Holm correction

^{*} Šidak single-step correction

^{**} Benjamini-Hochberg (false-discovery rate!)

2. Please enter the adjusted empirical P -values for markers rs1112 and rs1117 from the analysis with PLINK in the table below.

Marker	Empirical adjusted P -value (EMP2) after	
	5000 permutations	100000 permutations
rs1112		
rs1117		

3. The Bonferroni-adjusted P -value for rs1112 is, despite the conservativeness of this correction, much smaller than the adjusted empirical P -value. Do you have an explanation?

4. Please enter the raw and adjusted P -values for markers rs1112 and rs1117 from the analysis in R in the table below.

Marker	Raw P-value	Adjusted P -value following			
		Bonferroni	Holm	SidakSS [*]	BH ^{**}
rs1112					
rs1117					

^{*} Šidak single-step correction

^{**} Bonferroni-Holm (false-discovery rate!)

Answers

Multiple Testing

Correction with PLINK

Stepwise correction

```
plink --bfile dbp --assoc --adjust --out multtest
```

multtest.assoc.adjusted

CHR	SNP	UNADJ	GC	BONF	HOLM	SIDAK_SS	...	FDR_BH	...
11	rs1112	8.634e-09	0.0001088	1.727e-07	1.727e-07	1.727e-07	...	1.727e-07	...
11	rs1115	4.616e-07	0.000699	9.231e-06	8.77e-06	9.231e-06	...	4.616e-06	...
11	rs1117	1.654e-06	0.001273	3.308e-05	2.977e-05	3.308e-05	...	1.103e-05	...
11	rs1119	0.0001107	0.009339	0.002214	0.001882	0.002211	...	0.0005535	...
...									

The output file contains one line for each tested marker. The column ‘UNADJ’ contains the nominal P -value without correction for multiple testing. The subsequent columns contain the corrected P -values for different correction methods, e.g. ‘BONF’ for Bonferroni correction and ‘SIDAK_SS’ for Šidak correction.

Correction by permutation

```
plink --bfile dbp --assoc --mperm 5000 --out multperm5000
```

```
plink --bfile dbp --assoc --mperm 100000 --out multperm100000
```

multperm5000.assoc.mperm

CHR	SNP	EMP1	EMP2
11	rs1101	0.6727	1
11	rs1102	0.8414	1
...			
11	rs1112	0.0002	0.0002
11	rs1113	0.0004999	0.0022
...			
11	rs1117	0.0002	0.0002
...			

multperm100000.assoc.mperm

CHR	SNP	EMP1	EMP2
11	rs1101	0.6801	1
11	rs1102	0.8389	1
...			
11	rs1112	1e-05	1e-05
11	rs1113	0.000125	0.00216
...			
11	rs1117	1e-05	6e-05
...			

The column ‘EMP2’ contains the desired maxT (‘Westfall & Young’) empirical P -value, corrected for multiple testing (e.g. $p=6.0 \times 10^{-5}$ for rs1117). The ‘EMP1’ column contains the empirical P -value for the single-marker test *without* multiplicity correction (e.g. $p=1.0 \times 10^{-5}$ for rs1117) based on permutation rather than on asymptotic statistical theory, like the ξ^2 test.

Correction with R

```
load ("p.values.R")
```

```
ls ()
```

```
[1] p.values
```

p.values

```

rs1101      rs1102      rs1103      rs1104      rs1105
7.277672e-01 8.958136e-01 4.179383e-01 7.649468e-02 7.280132e-01
rs1106      rs1107      rs1108      rs1109      rs1110
```

```

6.979278e-01 4.219864e-01 1.838581e-01 1.000000e+00 6.428679e-01
rs1111 rs1112 rs1113 rs1114 rs1115
7.709398e-01 1.253122e-08 2.043651e-04 2.920328e-01 6.485844e-07
rs1116 rs1117 rs1118 rs1119 rs1120
9.152594e-01 2.395866e-06 3.379778e-04 1.497200e-04 4.326036e-04

```

Loading a dedicated R library

```
library(multtest)
```

```
Load required package: BiocGenerics
```

```
Load required package: parallel
```

```
...
```

```
Load required package: Biobase
```

```
Welcome to Bioconductor
```

```
Welcome to Bioconductor
```

```
...
```

Stepwise correction

```
adj.p.values = mt.rawp2adjp(p.values, c("Bonferroni", "Holm", "SidakSS", "BH"))
```

```
adj.p.values
```

```
$adjp
```

```

          rawp Bonferroni Holm SidakSS                      BH
[1,] 1.253122e-08 2.506245e-07 2.506245e-07 2.506244e-07 2.506245e-07
[2,] 6.485844e-07 1.297169e-05 1.232310e-05 1.297161e-05 6.485844e-06
[3,] 2.395866e-06 4.791732e-05 4.312558e-05 4.791622e-05 1.597244e-05
[4,] 1.497200e-04 2.994400e-03 2.545240e-03 2.990145e-03 7.486001e-04

```

```
...
```

```
$index
```

```
[1] 12 15 17 19 13 18 20 4 8 14 3 7 10 6 1 5 11 2 16 9
```

```
$h0.ABH
```

```
NULL
```

```
$h0.TSBH
```

```
NULL
```

The resulting data object is a list. The first list entry ('adjp') is a matrix of raw and multiple-testing adjusted P-values. These values are sorted in ascending order; correction is applied by multiplying the nominal P-value with some correction factor. For each requested correction method, a corresponding column is appended to the column of nominal ('raw') P-values. The second list entry ('index') reports the index with regard to the original P-value list, i.e. p.values.

We subsequently use the names of the p.values object to assign row names to the P-value matrix for better readability, i.e. which P-value belongs to which marker:

```
rownames(adj.p.values$adjp) = names(p.values[adj.p.values$index])
```

```
adj.p.values$adjp
```

```

          rawp      Bonferroni      Holm      SidakSS      BH
rs1112 1.253122e-08 2.506245e-07 2.506245e-07 2.506244e-07 2.506245e-07
rs1115 6.485844e-07 1.297169e-05 1.232310e-05 1.297161e-05 6.485844e-06
rs1117 2.395866e-06 4.791732e-05 4.312558e-05 4.791622e-05 1.597244e-05
rs1119 1.497200e-04 2.994400e-03 2.545240e-03 2.990145e-03 7.486001e-04
rs1113 2.043651e-04 4.087302e-03 3.269841e-03 4.079376e-03 8.174604e-04
rs1118 3.379778e-04 6.759556e-03 5.069667e-03 6.737897e-03 1.126593e-03
rs1120 4.326036e-04 8.652072e-03 6.056451e-03 8.616607e-03 1.236010e-03
rs1104 7.649468e-02 1.000000e+00 9.944309e-01 7.963952e-01 1.912367e-01
rs1108 1.838581e-01 1.000000e+00 1.000000e+00 9.828085e-01 4.085736e-01
rs1114 2.920328e-01 1.000000e+00 1.000000e+00 9.989994e-01 5.840657e-01
rs1103 4.179383e-01 1.000000e+00 1.000000e+00 9.999801e-01 7.033107e-01
rs1107 4.219864e-01 1.000000e+00 1.000000e+00 9.999827e-01 7.033107e-01
rs1110 6.428679e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.069880e-01
rs1106 6.979278e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.069880e-01
rs1101 7.277672e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.069880e-01
rs1105 7.280132e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.069880e-01

```

```
rs1111 7.709398e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.069880e-01
rs1102 8.958136e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.634309e-01
rs1116 9.152594e-01 1.000000e+00 1.000000e+00 1.000000e+00 9.634309e-01
rs1109 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
```

Questions

1. Please enter the raw and stepwise adjusted P -values for markers rs1112 and rs1117 from the PLINK analysis in the table below.

Marker	Raw P-value	Adjusted P -value following			
	UNADJ	BONF [±]	HOLM [¶]	SIDAK_SS [*]	FDR_BH ^{**}
rs1112	8.634e-09	1.727e-07	1.727e-07	1.727e-07	1.727e-07
rs1117	1.654e-06	3.308e-05	2.977e-05	3.308e-05	1.103e-05

± Bonferroni correction

¶ Holm correction

* Šidak single-step correction

** Benjamini-Hochberg (false-discovery rate!)

2. Please enter the adjusted empirical P -values for markers rs1112 and rs1117 from the PLINK analysis in the table below.

Marker	Empirical adjusted P -value (EMP2) after	
	5000 permutations	100000 permutations
rs1112	0.0002	1e-05
rs1117	0.0002	6e-05

3. The Bonferroni-adjusted P -value for rs1112 is, despite the conservativeness of this correction, much smaller than the adjusted empirical P -value. Do you have an explanation?

Empirical P -values are calculated as the proportion of permutations that yield an even more extreme value for the test statistic (maxT) than that observed with the original sample. With small P -values, the number of permutations acts as a “resolution” limit for estimating the empirical P -value, because extreme values are by definition rare events. The inverse of the number of permutations equals the smallest P -value larger than zero that can be obtained by permutation testing; here: $1/5000 = 0.0002$ and $1/100000 = 1e-05$. Correcting P -value that are known to be very small therefore requires a large number of permutations and, correspondingly, a considerable amount of time.

4. Please enter the raw and adjusted P -values for markers rs1112 and rs1117 from the R analysis in the table below.

Marker	Raw P-value	Adjusted P -value following			
		Bonferroni	Holm	SidakSS [*]	BH ^{**}
rs1112	1.253e-08	2.506e-07	2.506e-07	2.506e-07	2.506e-07
rs1117	2.396e-06	4.792e-05	4.313e-05	4.792e-05	1.597e-05

* Šidak single-step correction

** Bonferroni-Holm (false-discovery rate!)

Exercise

Multifactorial Analysis 2

Analyses using PLINK

Here, we continue the regression exercise and test for gene-gene and gene-environment interaction. Since the syntax for many of the commands is repetitive, please use the copy & paste functionality of your text editor and subsequently make the necessary changes to the copied text.

Attention: PLINK expects each command to be in a single line! PLINK ignores arguments on subsequent lines after a line break. Please type each command without a line break or use a backslash ('\') before a line break. A backslash causes PLINK to ignore the line break.

Please also answer the questions at the end of the exercise.

The data set

Please change the working directory as requested. You are provided with a data set on diastolic blood pressure and the genotypes of 20 SNP markers. The data are already in PLINK format. There are some files:

- **dbp.[fam|bim|bed]:** Set of binary PLINK files with a dichotomized trait (affection status: elevated blood pressure yes/no)

Use a text editor (notepad/Wordpad under Windows, pico/vi/nano/emacs under Linux) to inspect the contents of these files (except for *.bed file which is binary). Make sure you understand the meaning of each column in the files.

For this exercise, data cleaning will be skipped. First, please have a look to the questions sheet in the back. Enter the *P*-values in the table while proceeding with the exercise.

IV. Gene-environment (GxE) and gene-gene (GxG) interaction

Interaction between factors (genetic and non-genetic) can also be tested in regression models. The model then includes a main effect term for each factor as well as additional product terms for all pairs of factors. With PLINK, use the `--interaction` flag to include interaction terms in the model.

It is important to note, however, that statistical interaction does not necessarily imply biological interaction, such as epistasis or synergy. *Statistical interaction only denotes the deviation from linearity in the regression model!*

Gene-environment (GxE) interaction

Run a regression analysis where all SNPs are considered under an allelic model, where the effects are adjusted for the effect of the covariate sex, and where additionally the gene-environmental interaction of SNP marker and sex is considered:

```
plink --bfile dbp --logistic sex interaction --out logreg.sex.inter.add
```

Inspect the results file with a text editor:

```
logreg.sex.inter.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	1.124	0.636	0.5248
11	rs1101	1021	1	SEX	600	2.769	3.62	0.0002948
11	rs1101	1021	1	ADDxSEX	600	0.7912	-0.9564	0.3389
...								

The ADD and SEX lines contain the *P*-values for the marker and the environmental covariate, respectively. The ADD×SEX line gives the *P*-value for the interaction term.

Gene-gene (G×G) interaction

Gene-gene interaction is incorporated in a very similar fashion by combining the `--condition` and `--interaction` flags. Run a regression analysis as before, but now incorporate SNP-SNP interaction terms while adjusting for SNP rs1112:

```
plink --bfile dbp --logistic interaction --condition rs1112 \
      --out logreg.snp1112.inter.add
```

Inspect the results file with a text editor.

`logreg.snp1112.inter.add.assoc.logistic`

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
11	rs1101	1021	1	ADD	600	0.7121	-2.034	0.04195
11	rs1101	1021	1	rs1112	600	1.377	1.535	0.1247
11	rs1101	1021	1	ADD×CSNP1	600	1.721	2.686	0.007232
...								

The ADD and rs1112 lines contain the *P*-values for the considered marker and for SNP rs1112 as the factor the analysis is adjusted for, respectively. The ADD×CSNP1 line contains the *P*-value for the interaction term.

Questions

1. Please enter the *P*-values for marker rs1112 from the analyses in the table below.

	Type of analysis	<i>P</i> -value
IV.	Interaction <i>P</i> -value with covariate sex	
	Interaction <i>P</i> -value with marker rs1117	

2. Is there evidence for statistical interaction between marker rs1112 and sex?

3. Is there evidence for statistical interaction between markers rs1112 and rs1117?

Analyses using R

In this exercise, we continue the regression analysis with considering gene-gene and gene-environmental interaction as well as model selection.

The data set is the same as with the PLINK exercise. For convenience, it has already been converted to R format and stored in the file `dbp.R`.

Since the syntax for many of the commands is highly repetitive and in order to save time, please use the copy & paste functionality of your text editor and subsequently make the necessary changes to the copied text.

Please also answer the questions at the end of the exercise.

Data set import

Start R and change the working directory as requested. Load the data set for the exercise and get an overview which objects have been loaded into the R working memory:

```
load("dbp.R")
ls()
dbp[1:5,]
summary(dbp)
```

IV. Gene-environment (GxE) and gene-gene (GxG) interaction

Interaction between factors (genetic and non-genetic) can also be tested. The model then additionally includes the product term of the two factors. In R, this is achieved by using the `*` operator in the model formulation, for example `affection ~ sex * snp`, which is equivalent to `affection ~ sex + snp + sex:snp`. The variables `sex` and `snp` denote the main effect terms, while `sex:snp` denotes the interaction term.

It is important to note, however, that statistical interaction does not necessarily imply biological interaction, such as epistasis or synergy. *Statistical interaction only denotes the deviation from linearity within the regression model!*

Gene-environment (GxE) interaction

Test SNP `rs1112` for significant interaction with each of the two covariates `sex` and `age`:

```
result.inter = glm (affection ~ sex * rs1112, family=binomial("logit"),
                    data=snp.data)
summary(result.inter)

result.inter = glm (affection ~ age * rs1112, family=binomial("logit"),
                    data=snp.data)
summary(result.inter)
```

Gene-gene (GxG) interaction

Now test markers `rs1112` and `rs1117` for significant statistical interaction:

```
result.inter = glm (affection ~ rs1112 * rs1117, family=binomial("logit"),
                    data=snp.data)
summary(result.inter)
```

V. Model selection

The use of *parsimonious* statistical models to describe the relation between response and predictor variables is recommended. Model selection can often help to extract the relevant variables from a list of potential candidates. For example, one SNP among numerous ones in strong LD is sufficient to represent the underlying phenotypic association. Model selection offers a convenient way to select only those markers that show independent signals.

In a first step, run a logistic regression analysis of the full model, which includes *all* markers and *all* covariates under consideration:

```
result.reg = glm (affection ~ sex + age + rs1112 + rs1117,
                  family=binomial("logit"), data=snp.data)
summary(result.reg)
```

Now perform the model selection using R's `step` function. R will document the steps that led to the exclusion of variables on screen. Use the `summary` function to print the result of the model selection process:

```
modelchoice.result <- step (result.reg)
summary(modelchoice.result)
```

Quitting

Quit the R session by calling the quit function. Note that this exercise will be continued in part II, so please type Y to save your workspace image.

```
q()
```

➔ (Y)es for saving the workspace image.

Questions

1. Please enter the *P*-values for marker rs1112 from the analyses in the table below.

	Type of Analysis	<i>P</i> -value
IV.	Interaction <i>P</i> -value with covariate sex	
	Interaction <i>P</i> -value with marker rs1117	

2. Which variables are included in the final regression model after model selection?

Answers

Multifactorial Analysis 2

Analyses using PLINK

IV. Gene-environment (GxE) and gene-gene (GxG) interaction

```
plink --bfile dbp --logistic sex interaction --out logreg.sex.inter.add  
logreg.sex.inter.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1112	1245604	2	ADD	600	2.115	3.681	0.0002325
11	rs1112	1245604	2	SEX	600	2.204	3.361	0.0007774
11	rs1112	1245604	2	ADDxSEX	600	1.042	0.1491	0.8815
...								

```
plink --bfile dbp --logistic interaction --condition rs1112 \  
--out logreg.snpl112.inter.add  
logreg.snpl112.inter.add.assoc.logistic
```

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
...								
11	rs1117	1258119	2	ADD	600	1.451	0.8555	0.3923
11	rs1117	1258119	2	rs1112	600	1.846	3.125	0.001779
11	rs1117	1258119	2	ADDxCSNP1	600	0.9281	-0.2363	0.8132
...								

When considering the effects of a covariate and a possible interaction between marker and covariate, the output file contains *three* lines. One line reports the results for the tested marker ('ADD'), while a second line reports the results for the covariate (e.g. 'SEX'). A third line (e.g. 'ADDxSEX') reports the results for the interaction term between marker and covariate in the regression model. In this example, marker rs1112 shows significant

association with the affection status ($p=0.0002$) after correction for the effect of sex and a potential interaction between rs1112 and sex. While sex shows significant phenotypic association ($p=0.0008$), the interaction term is not significant ($p=0.9$). Thus, while males and females do have different baseline risks, there is no evidence that the risk *increase* caused by the minor allele (allele '2') differs between males and females.

Questions

1. Please enter the *P*-values for marker rs1112 from the analyses in the table below.

	Type of analysis	<i>P</i> -value
IV.	Interaction <i>P</i> -value with covariate sex	0.8815
	Interaction <i>P</i> -value with marker rs1117	0.8132

2. Is there evidence for statistical interaction between marker rs1112 and sex?

No, the *P*-value of 0.8815 for the interaction term in the logistic regression model does not indicate a deviation from linearity in the model.

3. Is there evidence for statistical interaction between markers rs1112 and rs1117?

No.

Analyses using R

IV. Gene-environment (GxE) and gene-gene (GxG) interaction

Gene-environment (GxE) interaction

```
# --- Interaction between sex and marker --- #
result.inter = glm (affection ~ sex * rs1112, family=binomial("logit"),
                    data=snp.data)

summary(result.inter)
Call:
glm(formula = affection ~ sex * rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8388  -1.1205  -0.0965   1.2176   1.5685

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.09415    0.15371  -0.613  0.540174
sex2         -0.79026    0.23515  -3.361  0.000777 ***
rs1112        0.79049    0.18896   4.183  2.87e-05 ***
sex2:rs1112  -0.04141    0.27771  -0.149  0.881472
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 774.96 on 596 degrees of freedom
AIC: 782.96

Number of Fisher Scoring iterations: 4
```

While marker rs1112 is associated with affection status after adjusting for the effect of sex (Wald test: $p=2.9 \times 10^{-5}$), there is no evidence that the marker risk allele causes different risk *increases* in females compared to males ($p=0.9$).

```
# --- Interaction between age and marker --- #
result.inter = glm (affection ~ age * rs1112, family=binomial("logit"),
                    data=snp.data)
summary(result.inter)
Call:
glm(formula = affection ~ age * rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8044  -1.0479  -0.1256   1.0606   1.4655

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.764365   0.328207  -2.329  0.01986 *
age           0.005719   0.005508   1.038  0.29909
rs1112        1.193715   0.393377   3.035  0.00241 **
age:rs1112    -0.007716   0.006585  -1.172  0.24130
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 796.26  on 596  degrees of freedom
AIC: 804.26
```

Number of Fisher Scoring iterations: 4

While marker rs1112 is associated with affection status after adjusting for the effect of age (Wald test: $p=2.9 \times 10^{-5}$), there is no evidence for a significant association of age with affection status ($p=0.3$) nor an interaction between age and the marker ($p=0.2$).

Gene-gene (GxG) interaction

```
# --- Interaction between markers rs1112 and rs1117 --- #
result.inter = glm (affection ~ rs1112 * rs1117, family=binomial("logit"),
                    data=snp.data)
summary(result.inter)
Call:
glm(formula = affection ~ rs1112 * rs1117, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7167  -0.9899  -0.1342   1.1126   1.3773

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.45855    0.11749  -3.903  9.5e-05 ***
rs1112        0.61285    0.19612   3.125  0.00178 **
rs1117        0.37232    0.43522   0.855  0.39228
rs1112:rs1117 -0.07464    0.31590  -0.236  0.81323
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 796.16 on 596 degrees of freedom
AIC: 804.16
Number of Fisher Scoring iterations: 4

While marker rs1112 is associated with affection status after adjusting for the effect of marker rs1117 (Wald test: $p=0.002$), there is no evidence for a significant association of marker rs1117 with affection status ($p=0.4$) nor an interaction between both markers ($p=0.8$), i.e. that the genotype of rs1117 may have a modifying effect on the risk increase caused by marker rs1112.

V. Model selection

```
# --- Regression analysis of full model (including all variables) --- #
result.reg = glm (affection ~ sex + age + rs1112 + rs1117,
                  family=binomial("logit"), data=snp.data)
summary(result.reg)
Call:
glm(formula = affection ~ sex + age + rs1112 + rs1117, family = binomial("logit"),
    data = snp.data)
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-1.9152	-1.1211	-0.1128	1.1820	1.6111

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.211378	0.264220	-0.800	0.42371
sex2	-0.808941	0.173019	-4.675	2.93e-06 ***
age	0.002183	0.004109	0.531	0.59519
rs1112	0.635080	0.193437	3.283	0.00103 **
rs1117	0.233915	0.235438	0.994	0.32045

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 773.74 on 595 degrees of freedom
AIC: 783.74

Number of Fisher Scoring iterations: 4

```
# --- Backward model selection, starting with full model --- #
modelchoice.result <- step (result.reg)
```

Start: AIC=783.74
affection ~ sex + age + rs1112 + rs1117

	Df	Deviance	AIC
- age	1	774.02	782.02
- rs1117	1	774.72	782.72
<none>		773.74	783.74
- rs1112	1	784.97	792.97
- sex	1	796.08	804.08

Step: AIC=782.02

affection ~ sex + rs1112 + rs1117

	Df	Deviance	AIC
- rs1117	1	774.98	780.98
<none>		774.02	782.02
- rs1112	1	785.34	791.34
- sex	1	796.21	802.21

```

Step:  AIC=780.98
affection ~ sex + rs1112
      Df Deviance    AIC
<none>      774.98 780.98
- sex      1    797.75 801.75
- rs1112   1    808.19 812.19

# --- Final model (result of selection procedure) --- #
summary(modelchoice.result)
Call:
glm(formula = affection ~ sex + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
      Min       1Q   Median       3Q      Max
-1.82645  -1.12415  -0.09007   1.21323   1.57462
Coefficients:
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.08386    0.13730  -0.611    0.541
sex2         -0.81412    0.17253  -4.719 2.37e-06 ***
rs1112        0.77139    0.13840   5.574 2.49e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
Null deviance: 831.78 on 599 degrees of freedom
Residual deviance: 774.98 on 597 degrees of freedom
AIC: 780.98

Number of Fisher Scoring iterations: 4

```

The model selection procedure started with the full model, i.e. it contained all predictor terms that were to be considered. If interaction terms should also be considered, these would have to be included in the initial model estimation request (i.e. the `glm` call). Covariates age and rs1117 were subsequently discarded from the model since there do not significantly improve the model fit (i.e. decrease the error) or, with backward selection, did not significantly worsened the model fit (i.e. dropping did not lead to largely increased error). Model fit is measured by the AIC criterion which, in addition to the deviance, penalized larger numbers of predictors in the regression model. The final model includes sex and rs1112 as predictors.

Questions

1. Please enter the *P*-values for marker rs1112 from the analyses in the table below.

	Type of Analysis	<i>P</i> -value
IV.	Interaction <i>P</i> -value with covariate sex	0.881472
	Interaction <i>P</i> -value with marker rs1117	0.81323

2. Which variables are included in the final regression model after model selection?

sex and marker rs1112