Regression

In this exercise, we'll carry out a number of logistic and linear regression analyses to test the association of several SNPs with an affection status and with a quantitative trait. This includes the use of different tests, the calculation of odds ratios (OR), and the consideration of different genetic models. Further objectives are adjusting for the effects of covariates, and testing a SNP for association given the effect of another SNP. Finally, we'll consider gene-gene and gene-environment interaction, as well as model selection.

Data overview

This uses the same diastolic blood pressure (DBP) data set as the PLINK/R introduction exercise: 600 individuals genotyped at 20 SNP markers on chromosome 11, along with sex, age, and both a quantitative and a dichotomized version of the trait. The files used in this notebook:

- `dbp.cc.ped` / `dbp.map` — pedigree and map files with family, sex, the dichotomized trait (affected yes/no), and genotypes; used for the case-control (logistic regression) analyses in Part 1.
- `dbp.qt.ped` / `dbp.map` — the same individuals and genotypes, but with the original quantitative trait value instead of the dichotomized one; used for the linear regression analysis in Part 1.
- `dbp.age.pheno` — covariate file containing the age of each individual.
- `dbp.bed` / `dbp.bim` / `dbp.fam` — the case-control data in PLINK's binary format, used for the multiple-testing section (`--assoc --adjust` / `--mperm`).
- `dbp.R` — an R data file containing the same information as a ready-to-use data frame, `dbp` (600 rows × 28 columns: family/individual IDs, `sex` , `trait` , `affection` , `age` , and genotypes for all 20 markers, `rs1101` – `rs1120`). This is what Part 2 loads and works with directly, instead of parsing PLINK files — most of the exercise focuses on two of the markers, `rs1112` and `rs1117` .
- `p.values.R` — a small, separate set of p-values used only in the multiple-testing section of Part 2, to demonstrate different correction methods.

References

- Chang CC, Chow CC, Tellier LCAM, Vattikuti S, Purcell SM, Lee JJ (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience, 4.
- Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC (2007) PLINK: a toolset for whole-genome

association and population-based linkage analysis. *American Journal of Human Genetics*, 81:559-575.
- R Core Team (2025), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing

Part 1: PLINK

> **Kernel:** run the code cells in this section with the **Bash** kernel.

In [ ]:
```
# Set the output directory for this exercise
output_dir="./output/regression"
```

In [ ]:
```
# I. Logistic regression on a single SNP under an allelic model
plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.add.ci.assoc.logistic
```

In [ ]:
```
# II. Adjustment for the effects of covariates and of other SNPs
plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.age.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.sex.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.sexage.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.snp1112.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.snp1117.add.assoc.logistic
```

In [ ]:
```
# III. Analysis of quantitative instead of dichotomized trait
plink --ped ./data/dbp.qt.ped --map ./data/dbp.map --out ${output_dir}/linre
cat ${output_dir}/linreg.sex.add.assoc.linear
```

In [ ]:
```
# IV. Gene-environment (GxE) and gene-gene (GxG) interaction
plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.sex.inter.add.assoc.logistic

plink --ped ./data/dbp.cc.ped --map ./data/dbp.map --out ${output_dir}/logre
cat ${output_dir}/logreg.snp1112.inter.add.assoc.logistic
```

In [ ]:
```
# V. Multiple testing

plink  --bfile ./data/dbp  --assoc  --adjust  --out ${output_dir}/multtest
```

```
plink  --bfile ./data/dbp  --assoc  --mperm   5000  --out ${output_dir}/mult
plink  --bfile ./data/dbp  --assoc  --mperm 100000  --out ${output_dir}/mult
```

Part 2: R

> **Kernel:** run the code cells in this section with the **R** kernel.

In [ ]:
```
# Load the data set for the exercise
load ("./data/dbp.R")
ls ()
dbp[1:5,]

# Get an overview which objects have been loaded into the R working memory
summary(dbp)
```

I. Logistic Regression on a Single SNP Genotype

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

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

In [ ]:
```
# Regression + Wald test

# Assign the results from the regression analysis tothe new object
result.snp12 = glm (affection ~ rs1112, family=binomial("logit"), data=dbp)
# Print the results
print ( result.snp12 )

# Membership in these classes causes R to use dedicated, specialized functic
print ( class (result.snp12) )
print ( summary(result.snp12) )
```

The marker variable `rs1112` is of data type factor (nominal), so we've considered a general genotypic model. R has therefore created two dummy variables, `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 belongs to the special R classes `lm` and `glm` (same names as the functions).

The coefficients table lists the estimated regression coefficients (β) along with their standard errors, and the P-values from a Wald test.

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

In [ ]:
```
# LRT #
dev.geno = anova (result.snp12, test="Chi")
  lrt.pvalue = pchisq(dev.geno[dim(dev.geno)[1],"Deviance"], df=2, ncp=0, FA
  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^β), as well as their confidence intervals.

In [ ]:
```
# OR + CI #
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've considered a general genotypic model. For an allelic (multiplicative) model, the data type has to be changed to numeric. This recodes the genotype from nominal 2/3/4 (for 11/12/22) to numeric 0/1/2, representing the number of copies of the "2" allele carried by each sample.

In [ ]:
```
# Allelic model #
snp.data = dbp[,c("affection", "rs1112")]
  summary(snp.data)
snp.data[,"rs1112"] <- as.numeric(snp.data[,"rs1112"]) - 1
  summary(snp.data)

# Results for allelic model (allele 2) #
result.all = glm (affection ~ rs1112, family=binomial("logit"), data=snp.dat
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 adjusting for their effect by simply incorporating them into the statistical model.

In [ ]:
```
# --- Excerpt for subsequent analysis --- #
snp.data = dbp[,c("affection", "trait","sex", "age", "rs1112", "rs1117")]
  summary(snp.data)
```

```
# Allelic model will be considered as the model for the markers #
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`, first with an adjustment for sex, then for age (which is often suspected to influence the trait of interest), and finally for both covariates, sex and age, simultaneously.

In [ ]:
```
# Covariates
# Adjust for sex
result.adj = glm (affection ~ sex + rs1112      , family=binomial("logit"),
  summary(result.adj)

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

# Adjust for sex & age
result.adj = glm (affection ~ sex + age + rs1112, family=binomial("logit"),
  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 whether a newly found association is independent of such established ones — equivalent to adjusting for the effect of the already established SNP. The section below runs a logistic regression analysis for each of the two SNPs `rs1112` and `rs1117`, while adjusting for the effect of the other.

Note that the P-values from a Wald test don't differ between the two orderings of markers, but the P-values from a likelihood-ratio test (obtained from the `anova` function) do.

In [ ]:
```
# Effect of SNP2 given SNP 1 #
result.adj = glm (affection ~ rs1117 + rs1112, family=binomial("logit"), dat
  summary(result.adj)
  anova (result.adj, test="Chi")
result.adj = glm (affection ~ rs1112 + rs1117, family=binomial("logit"), dat
  summary(result.adj)
  anova (result.adj, test="Chi")
```

III. Analysis of Quantitative Instead of Dichotomized Trait

Dichotomizing quantitative trait values can result in a loss of power, because information is discarded. In our example data set, the original trait value (diastolic blood pressure) had been dichotomized into case-control status: individuals with a value above a certain threshold were defined as having high blood pressure ("cases"), while the rest were considered controls with normal blood pressure.

The column `trait` in the data frame `dbp` contains the original quantitative trait values. This section runs two linear regression analyses, one without and one with adjustment for the effect of sex.

In [ ]:
```
# Consideration of QT instead of binary affection status (dichotomized QT) #
result.adj = lm (trait ~ rs1112      , data=snp.data)
  summary(result.adj)
result.adj = lm (trait ~ sex + rs1112, data=snp.data)
  summary(result.adj)
```

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 with the `*` operator in the model formula — for example, `affection ~ sex * snp` is equivalent to `affection ~ sex + snp + sex:snp`. Here, `sex` and `snp` are the main effect terms, while `sex:snp` is the interaction term.

It's important to note, however, that statistical interaction does not necessarily imply biological interaction, such as epistasis or synergy — statistical interaction only denotes deviation from linearity within the regression model.

In [ ]:
```
# GxE : Test SNP rs1112 for significant interaction with each of the two cov
result.inter = glm (affection ~ sex * rs1112, family=binomial("logit"), data
  summary(result.inter)
result.inter = glm (affection ~ age * rs1112, family=binomial("logit"), data
  summary(result.inter)
```

In [ ]:
```
# GxG : Test markers rs1112 and rs1117 for significant statistical interacti
result.inter = glm (affection ~ rs1112 * rs1117, family=binomial("logit"), d
  summary(result.inter)
```

V. Model Selection

The models considered so far included covariates and SNPs chosen ad hoc, based on the earlier sections of this exercise. In practice, when several covariates and/or SNPs are candidates for a model, it helps to have a systematic procedure for deciding which terms belong in the final model — this is called **model selection**.

A common automated approach in R is stepwise selection based on the **Akaike Information Criterion (AIC)**, implemented by the `step()` function. AIC balances

model fit (the deviance — how well the model explains the data) against model complexity (the number of parameters), so that adding a term is only "worth it" if it improves the fit enough to offset the added complexity. Lower AIC indicates a better fit-vs-complexity trade-off.

Starting from the full model fit below (`affection ~ sex + age + rs1112 + rs1117`), `step()` repeatedly considers dropping or adding a term and compares the resulting AIC values; at each iteration it makes whichever single change reduces the AIC the most, and stops once no further change would lower it. By default (no `scope` argument supplied), `step()` searches in **both directions** between the intercept-only model and the model you passed in, so terms can be dropped and later re-added over the course of the search. The verbose output printed below shows, at each step, the candidate models under consideration (`- term` for removal, `+ term` for addition) ranked by resulting AIC; the model shown at the end is the one `step()` settled on.

A word of caution: stepwise selection is a useful exploratory tool, but it has well-known statistical limitations. It can be unstable — small changes in the data can lead to a different selected model — and the p-values in the final model no longer have their usual interpretation, because the same data were used both to choose the model and to test it (a form of implicit multiple testing that `step()` does not correct for). This matters especially in a genetic association study: if your goal is to test a specific SNP (e.g. `rs1112`) for association, be careful about reading its effect estimate off the AIC-selected model. If `step()` drops that SNP, that is itself an informative result (the term did not improve the model), but you should still report the SNP's effect from the pre-specified model that includes it (e.g. `result.reg` below), not chase the "best AIC" model as if it were the definitive answer.

If a more conservative selection is desired, `step(result.reg, k = log(nobs(result.reg)))` uses the **Bayesian Information Criterion (BIC)** instead of AIC — BIC penalizes additional parameters more heavily as the sample size grows, so it tends to favor smaller models than AIC.

In [ ]:
```
# --- V. Model selection --- #
result.reg = glm (affection ~ sex + age + rs1112 + rs1117, family=binomial("
  summary(result.reg)

modelchoice.result <- step (result.reg)
  summary(modelchoice.result)
```

In [ ]:
```
# VI. Multiple testing
## Load the library "multtest".
library (multtest)
# Load p values from data file
load("./data/p.values.R")
ls()
p.values
```

```
adj.p.values = mt.rawp2adjp(p.values,c("Bonferroni","Holm","SidakSS","BH"))
adj.p.values

rownames(adj.p.values$adjp) = names(p.values[adj.p.values$index])

adj.p.values$adjp
```

Questions

(Questions below are from the accompanying exercise sheet by Michael Nothnagel. Answer them using the results from the analyses you ran above.)

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, 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 `rs1117`	Wald: LRT:
III.	Single marker, quantitative trait, adjustment for sex	
IV.	Interaction P-value with covariate sex	
	Interaction P-value with marker `rs1117`	

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

Genotype	OR	95% CI
het (1/2)		–
hom (2/2)		–

3. 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?