# Quantitaive trait mixed model GWAS

Tamar Sofer

July 2017

# Quantitative trait analysis

- Quantitative trait = continuous outcome. The simplest to analyze.
- The basic linear regression model for a quantitative outcome:

$$y_i = \mathbf{x}_i^T \beta + g_i \alpha + \epsilon_i, i = 1, \ldots, n.$$

where here:

- $y_i$ is the trait value of person $i$.
- $\mathbf{x}_i$ is a vector of adjusting covariates (age, sex, etc.), $\beta$ is a vector of their effects.
- $g_i$ is the genotype dosage or count of the SNP (variant) of interes, $\alpha$ its effect.
- $\epsilon_i$ is a residual.

# Quantitative trait analysis

$$y_i = \mathbf{x}_i^T \beta + g_i \alpha + \epsilon_i, i = 1, \ldots, n.$$

- The basic assumption in this linear model is that observations are "independent and identically distributed" (i.i.d.).
- This does not hold for the HCHS/SOL.
  - So we cannot use the "usual" linear regression.
  - We use mixed models (or GEEs), instead.

Questions:

1. What will happen if we used linear regression, i.e. assume, contrary to fact, that participants are i.i.d?
2. How can we use linear regression correctly, if we really wanted to?

# Quantitative trait analysis

- The linear mixed model states that the traits of people who are somehow close or similar to each other, are more similar to each other than the trait values of people who are not close or similar.
  - I.e. some people's traits are correlated to each other.
- One way to model these correlations is using random effects.
- For example, if there was one source of such correlations:

$$y_i = \mathbf{x}_i^T \beta + g_i \alpha + b_i + \epsilon_i, i = 1, \ldots, n,$$

- with $b_i$ a random error - or a random effect - in addition to the i.i.d. errors $\epsilon_i$.

# Quantitative trait analysis

- Random effects model the correlation between individuals' trait values.
- Specifically, one can define a matrix to do that. E.g. a kinship matrix. Or a household matrix!

$$\operatorname{cor}\left[(b_1, b_2, b_3, \ldots)\right] = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & p_3 & \ldots \end{array} \\ \left(\begin{array}{cccc} 1 & 0 & 0.5 & \ldots \\ 0 & 1 & 0.5 & \ldots \\ 0.5 & 0.5 & 1 & \ldots \\ \vdots & & & \end{array}\right) \end{array}$$

- Here, the correlation between the random effects of persons $p_1$ and $p_2$ is 0, and that of $p_1$ and $p_3$ is 0.5. Etc.

# Linear mixed models

- Linear mixed models are similar to linear regression, with the addition of correlation information between peoples' traits.
- In the HCHS/SOL, we have three correlation matrices: kinship (also called genetic relatedness matrix, GRM), household, and block unit.
- The kinship matrix was estimated based on the genotyping data (using common variants, MAF$\geq 0.05$).
- The household and block unit matrices were calculated based on who people lived with in the same house, or block unit.

# Linear mixed models

So how are these correlation matrices used?

- While the correlation structures (the matrices) are pre-defined, the variance components are not.
- In linear regression (i.i.d. observations) there is a single residual variance: $\sigma_e^2$.
  - It is the variance of the i.i.d residuals: $\text{var}(\epsilon_i) = \sigma_e^2$.
  - In other words: a single variance component.
- In mixed models, there are at least 2 variance components. One for the i.i.d. errors, others correspond to random effects.
  - In the HCHS/SOL: $\sigma^2 = \sigma_e^2 + \sigma_g^2 + \sigma_h^2 + \sigma_c^2$

# Linear mixed models - variance components

- Variance components are used in two important applications.
  - Association testing;
  - Heritability estimation.
- Both require estimates of the variance components.
- They are estimated by fitting a null model.
  - A model that includes the trait, and all adjusting covariates, and the random effects matrices; but not individual genotypes.

# Linear mixed models - the null model

Let's try it!

- We first load our scanAnnotation object.

```
library(GWASTools)
library(GENESIS)
dir <- paste0("/home/postdoc/tsofer/SISG/",
    "Preparing_simulated_data_2")

scanAnnot <- getobj(file.path(dir,
                              "SISG_phenotypes.RData"))
scanAnnot
```

```
## An object of class 'ScanAnnotationDataFrame'
##   scans: 1 2 ... 500 (500 total)
##   varLabels: scanID EV1 ... group (8 total)
##   varMetadata: labelDescription
```

# Linear mixed models - the null model

- Select outcome, covariates, and load correlation matrices.

```r
varLabels(scanAnnot)[1:4]
```

```
## [1] "scanID" "EV1"    "EV2"    "sex"
```

```r
varLabels(scanAnnot)[5:8]
```

```
## [1] "age"     "trait"   "disease" "group"
```

```r
covariates <- c("EV1", "EV2", "sex", "age", "group")
outcome <- "trait"
HH.mat <- getobj(file.path(dir,
                  "SISG_houshold_matrix.RData"))
kin.mat <- getobj(file.path(dir,
                  "SISG_relatedness_matrix.RData"))
covMatList <- list(HH = HH.mat, kinship = kin.mat)
```

# Linear mixed models - the null model

```r
nullmod <- fitNullMM(scanData = scanAnnot,
                     outcome = outcome, covars = covariates,
                     covMatList = covMatList, verbose = FALSE)
```

# Linear mixed models - the null model

- Let’s look at the results:

```
names(nullmod)
```

```
##  [1] "varComp"              "varCompCov"           "fixef"
##  [4] "betaCov"              "fitted.values"        "resid.marg
##  [7] "eta"                  "resid.conditional"  "logLikR"
## [10] "logLik"               "AIC"                  "RSS"
## [13] "workingY"             "model.matrix"         "cholSigmaI
## [16] "scanID"               "family"               "converged"
## [19] "zeroFLAG"             "hetResid"
```

```
nullmod$varComp
```

```
##       V_HH V_kinship      V_E
##     0.0000    0.0000 231.7541
```

# Linear mixed models - the null model

- Let's look at the results:

```
nullmod$fixef
```

```
##                     Est         SE        Stat         pva
## (Intercept)  4.213919 2.47521363    2.898325 8.867166e-0
## EV1          5.532397 0.68118649   65.962115 4.596743e-1
## EV2         -3.191225 0.69292155   21.210297 4.115474e-0
## sexM         6.636157 1.37220159   23.388236 1.323857e-0
## age          3.771601 0.04967911 5763.733606 0.000000e+0
## groupuw     -4.847355 1.39223675   12.122256 4.982359e-0
```

# Linear mixed models - the null model

- Our simulated trait "trait" unfortunately doesn't seem to be very heritable.
- Let's simulate another outcomes to make it more interesting...

```
require(mvtnorm)
n <- nrow(kin.mat)
new.trait <-
  nullmod$model.matrix %*% matrix(c(4, 5, -2, 1, 4,2)) +
    matrix(rmvnorm(n = 1, mean = rep(0, n),
            sigma = diag(rep(120, n)) +
                      80*kin.mat + 40*HH.mat))
scanAnnot$new.trait <- as.numeric(new.trait)
```

# Linear mixed models - the null model

```
set.seed(101)
nullmod <- fitNullMM(scanData = scanAnnot,
              outcome = "new.trait",
              covars = covariates,
              covMatList = covMatList, verbose = FALSE)

nullmod$varComp
```

```
##       V_HH  V_kinship        V_E
##  49.773211   9.176902 164.682650
```

```
varCompCI(nullmod, prop = TRUE)
```

```
##           Proportion   Lower 95  Upper 95
## V_HH      0.22256672 -0.2742835 0.7194169
## V_kinship 0.04103559 -0.9556135 1.0376847
## V_E       0.73639769 -0.3309738 1.8037691
```

# The linear mixed model and heritability

- The proportion of variance due to kinship/genetic relatedness is heritability.
  - AKA narrow-sense heritability.
  - The heritability of "trait" is estimated to be 4%, with 95% confidence interval (-96, 104)%.
  - To test heritability we can use the confidence intervals - if they are calculated correctly(!), or the likelihood ratio test.
- The simulated data set has 500 people, which is very small.
- Therefore, variance components are not well estimated,
- and the confidence interval of the heritabability includes impossible values.
  - Negative values, and larger than 100...
  - There are methods to calculate feasible CIs.

# The linear mixed model and association testing

- After estimating variance components in the “null model", they are assumed fixed.
- We now use this null model object in association testing.
  - Note: it can take a long time to estimate variance components, so doing it once (instead of separately for every genetic variant) saves a lot of time.

```
gds <- GdsGenotypeReader(file.path(dir,
                                   "SISG_snp_dosages.gds"))
# assoc <- assocTestMM(genoData = gds,
#                      nullMMobj = nullmod)
# try to run! it'll give an error.
```

# The linear mixed model and association testing

```r
snpAnnot <- getobj(file.path(dir,
                    "SISG_snp_dosages_snpAnnot.RData"))
genoData <- GenotypeData(gds,
                  snpAnnot=snpAnnot, scanAnnot = scanAnnot)
assoc <- assocTestMM(genoData = genoData,
                     nullMMobj = nullmod)
```

```
## Running analysis with 500 Samples and 7463 SNPs

## Beginning Calculations...

## Block 1 of 2 Completed - 1.6 secs

## Block 2 of 2 Completed - 0.9335 secs
```

# The linear mixed model and association testing

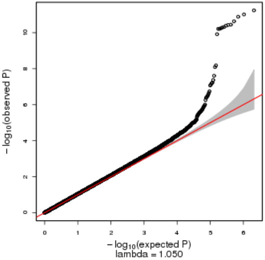
```r
head(assoc)
```

```
##    snpID chr   n   MAF minor.allele        Est        SE
## 1     1   1 500 0.000            A         NA        NA
## 2     2   1 500 0.001            A 16.5313629 15.056120
## 3     3   1 500 0.008            A  2.0362520  5.334783
## 4     4   1 500 0.000            A         NA        NA
## 5     5   1 500 0.209            B -0.1615227  1.139900
## 6     6   1 500 0.174            B -0.1977731  1.219728
##    Wald.pval
## 1        NA
## 2 0.2722119
## 3 0.7026887
## 4        NA
## 5 0.8873178
## 6 0.8711915
```

```r
close(gds)
```

# Inflation in Genome-Wide Association Studies

- Fundamental assumption in GWAS:
  - Most genetic variants are not associated with the outcome.
- Test statistics are mostly distributed “under the null"
- $\lambda_{gc} = \dfrac{\text{median(observed test statistics)}}{\text{median(expected distribution of test statistics)}}$
  - Ideally, $\lambda = 1$.
  - Because the bulk of the association are null.
- q-q plots are used to evaluate inflation.



# Exercises

1. Use the results from the GWAS that we ran on slide 18, and the function qqPlot() from the GWASTools package to make a q-q plot of the *p*-values from the Wald test.
2. What is the inflation factor? use the R code `pchisq(0.5, df = 1, lower.tail = FALSE)` to obtain the median of the expected distribution of the test statistics, and `median(assoc$Wald.Stat, na.rm = TRUE)` to obtain the median of the observed test statistics.
3. Can you evaluate whether the GWAS is too inflated or deflated?
4. Use the function manhattanPlot() from the GWASTools package to make a Manhattan plots for these *p*-values.

# Exercises

5. Set the significance threshold line in the Manhattan plot to be 0.05 divided by the number of tested variants (Bonferroni correction).
6. Show results in the Manhattan plot only for variants with imputation quality ("info") at least 0.8, or genotyped.
7. Which variant is most associated with "trait" among all variants (according to *p*-value)?
8. Use the parameter snp.include of function assocTestMM() to test only variants in positions 1029889 - 2136826 on chromosome 1.
   - Which variant has the most significant *p*-value?
9. Use the parameter scan.include of function fitNullMM() to perform association testing only in people from the UW group.
   - Is the most significant variant the same as before?