Disease trait mixed model GWAS

Tamar Sofer

July 2017

Disease (binary) trait analysis

- Disease trait - binary outcome, modeled as $D = 1$ (diseased) or $D = 0$ (disease free).
- The basic logistic regression model for a binary outcome:

$$\text{logit}[p(D_i = 1)] = \mathbf{x}_i^T \beta + g_i \alpha, i = 1, \dots, n.$$

where here:

  - D_i is the disease status of person i
  - $\mathbf{x}_i$ is a vector of adjusting covariates (age, sex, etc.), β is a vector of their effects.
  - g_i is the dosage or count of the genotype allele of interest.
  - $\text{logit}(u) = log[u/(1-u)]$, is a function that ensures that estimated disease probabilities - u - will always be in the range $(0, 1)$ (while $\text{logit}(u)$ could be anything).

Note: there is no "residual". In linear regression, the residual induces the variability. Here, we directly model a probability, which induces a variability.

Disease (binary) trait analysis

$$logit[p(D_i = 1)] = \mathbf{x}_i^T \beta + g_i \alpha, i = 1, \dots, n.$$

- The basic assumption in this 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" logistic regression.
  - We use mixed models (or GEEs), instead.

Questions:

1. What will happen if we used logistic regression instead of a logistic mixed model?
2. How can we use logistic regression correctly, assuming we really wanted to?

Disease (binary) trait analysis

- The logistic mixed model states that the disease probabilities of people who are somehow close or similar to each other, are more similar to each other than the disease probabilities of people who are not close or similar.
- One way to model this is using random effects.
- For example, if there was one source of such similarities between disease probabilities:

$$logit[p(D_i = 1)] = \mathbf{x}_i^T \beta + g_i \alpha + b_i, i = 1, \ldots, n,$$

- with b_i a random effect, increasing/decreasing the baseline odds of the disease.

Disease (binary) trait analysis

- Random effects reflect here similarity in disease odds across individuals using a correlation structure.
- As in linear regression, we use matrices to model the correlations between random effects across individuals.

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

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

Logistic mixed models

- Logistic mixed models are similar to logistic regression, with the addition of random effects.
- The interpretation is not as "clean" and simple as in linear regression.
  - In linear regression, we used random effects to explicitly model correlation between phenotypes across individuals.
  - Here, we do NOT explicitly model the equivalent - correlation between disease probabilities.

Logistic mixed models

So how are logistic mixed models practically different from linear mixed models?

- Variance components are still estimated (but no variance term corresponding to independent errors).
- There is no straight-forward interpretation of heritability based on variance components.
- Computationally, logistic models, and logistic mixed models, take longer to fit (compared to their linear counterparts).
- Logistic mixed models for more than a single correlation matrix are implemented in the software GMMAT and R package GENESIS (the same algorithm).

Logistic mixed models

- The GMMAT algorithm uses an approximation, which essentially fits linear mixed models about 4 times, each time for a different "working trait", until both "working traits" and estimated model parameters converge (become about the same as in the previous iteration).
- We still fit a null model, as in linear regression.
  - It takes about four times longer.
- We use the null model and the "working traits" to test genotype-disease associations.

Take-home message: the “null model” for binary traits takes 4 times longer to fit than that for quantitative traits. Afterwards computation time is the same.

Logistic vs linear mixed models

- In the past, people used linear mixed models instead of logistic mixed models.
  - Because it saved a lot of computation time.
- Is it okay to use linear mixed models?
  - Sometimes. But better not!
  - Basic assumption made by linear mixed models: residual variance is the same for all people.
  - Basic assumption made by logistic model: if someone has a probability p of disease, the variance of her outcome is $p(1-p)$.

Logistic vs linear mixed models

- Chen et al. (2016, AJHG) showed in “Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models" that when
  - MAF differ between sub-populations in the study
  - Disease prevalence differ between these sub-populations
- LMM test statistics can be either too significant (inflated), or too conservative (deflated).

Logistic vs linear mixed models

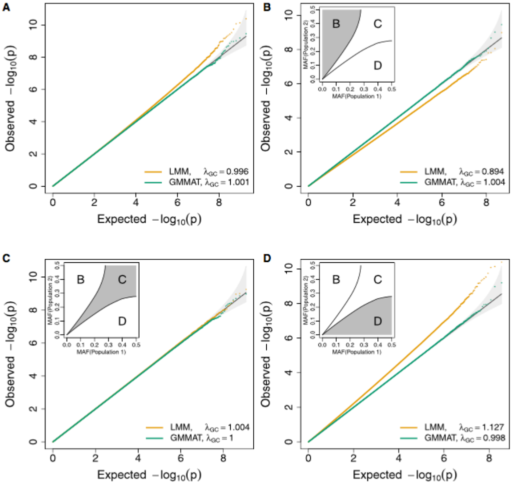

Figure 3. A Simulated Cohort Study with 10,000 Related Individuals
Quantile-quantile plots of association test p values from 3,200 simulation replicates under the null hypothesis of no genetic association, each with 625,583 common SNPs, were combined to get more than 2 billion null p values.
(A) All SNPs.
(B) Category 1: SNPs with the ratio of expected variances in population 1 (high risk) over population 2 (low risk) less than 0.8.
(C) Category 2: SNPs with the ratio of expected variances in population 1 (high risk) over population 2 (low risk) between 0.8 and 1.25.
(D) Category 3: SNPs with the ratio of expected variances in population 1 (high risk) over population 2 (low risk) greater than 1.25.

Logistic mixed models - the null model

Let's try it!

- We first load our scanAnnotation object.

```r
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.

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

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

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

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

```
covariates <- c("EV1", "EV2", "sex", "age", "group")
outcome <- "disease"
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,
                     family = "binomial", 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
## 0.1401686 0.0000000
```

Logistic mixed models - the null model

- Let's look at the results:

```
nullmod$fixef
```

```
##                     Est         SE        Stat        p
## (Intercept) -13.7912429 1.28407528 115.3521900 6.589170e
## EV1           0.6019640 0.14250266  17.8441160 2.397595e
## EV2          -0.4123798 0.15120786   7.4378266 6.386697e
## sexM          0.2579077 0.27949025   0.8515212 3.561224e
## age           0.2287610 0.02226677 105.5478697 9.263288e
## groupuw       0.1138958 0.29014617   0.1540927 6.946546e
```

The logstic 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.

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

The logistic mixed model and association testing

```
snpAnnot <- getobj(file.path(dir,
                    "SISG_snp_dosages_snpAnnot.RData"))
genoData <- GenotypeData(gds,
              snpAnnot=snpAnnot, scanAnnot = scanAnnot)
#assoc <- assocTestMM(genoData = genoData,
#                       nullMMobj = nullmod)
#
# try to run! it'll give an error.
```

The logistic mixed model and association testing

- We cannot use a Wald test for logistic mixed models
  - Wald test requires estimating genotype effects. In logistic regression, this requires re-estimation of variance components, impossible to do efficiently.
- Score tests are "under the null", so they are realistic for GWAS based on logistic mixed models.

```
assoc <- assocTestMM(genoData = genoData,
                     nullMMobj = nullmod,
                     test = "Score")
```

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

## Beginning Calculations...

## Block 1 of 2 Completed - 1.592 secs

## Block 2 of 2 Completed - 0.581 secs
```

The logistic mixed model and association testing

```
head(assoc)
```

```
##   snpID chr   n   MAF minor.allele         Score
## 1     1   1 500 0.000            A            NA
## 2     2   1 500 0.001            A -9.466802e-05 9.46489
## 3     3   1 500 0.008            A -6.244369e-01 6.70695
## 4     4   1 500 0.000            A            NA
## 5     5   1 500 0.209            B -4.808782e+00 1.19468
## 6     6   1 500 0.174            B -6.504122e+00 1.08241
##   Score.pval
## 1         NA
## 2 0.99223612
## 3 0.44577639
## 4         NA
## 5 0.16414719
## 6 0.04804926
```

```
close(gds)
```

Exercises

1. Use the results from the GWAS we ran on slide 19. Use the function qqPlot() from the GWASTools package to make a q-q plot figure of the *p*-values from the Score test.
2. Use the following approximation between the Score and Wald test to obtain log(ORs) and ORs for the SNP effects:
   - $\beta = \frac{\text{score}}{\text{var(score)}}$
   - $\text{SE}(\beta) = 1/\sqrt{\text{var(score)}}$
   - $\text{OR} = exp(\beta)$.

   - Which SNP has the highest odds ratio (OR)?
3. Use the function manhattanPlot() from the GWASTools package to generate a Manhattan plots for the Score test *p*-values. Did any of the SNPs achieve genome-wide significance? (*p*-value$= 5 \times 10{-}8$) array-wide significance?

Exercises

4. Which variant is most associated with "disease" among all variants?
5. Run linear mixed model GWAS instead of logistic, treating "disease" as a quantitative trait.
    - ...by first fitting a new null model using fitNullMM()
    - Compare the *p*-values obtained by the two methods. You can use a scatter plot or a q-q plot.
    - Do the two GWAS have the same top SNPs?
6. Use the parameter scan.include of function fitNullMM to fit perform association testing only in people from the UNC group.

    - Does the this GWAS have the same top SNP as the GWAS that was run on all participants?