Dealing with heterogeneity: group-specific variances and stratified analyses

Tamar Sofer

July 2017

The HCHS/SOL population is quite heterogeneous

1. Due to admixture:
   - Hispanics are admixed with three ancestral populations - European, African, Amerindian.
   - The patterns/proportions of admixture differ between background groups;
   - Which could be divided to Mainland: Mexican, Central American, South American, and Caribbean: Dominican, Cuban, and Puerto Rican.
2. Due to lifestyle and other environmental exposure differences.
3. Both genetic and environmental differences translate to differences in phentypic variability, disease prevalence.

The HCHS/SOL population is quite heterogeneous

Example 1: the prevalence of asthma in a subset of unrelated individuals, who live in different households.

Group	Participants	Current asthma
CentralAmerican	773	29 (3.8%)
SouthAmerican	499	22 (4.4%)
Mexican	2688	119 (4.4%)
PuertoRican	1298	339 (26.1%)
Cuban	1234	136 (11%)
Dominican	670	70 (10.4%)
All combined	7162	715 (10%)

The HCHS/SOL population is quite heterogeneous

Example 2: mean and variance of height in a subset of unrelated individuals, who live in different households.

Group	Participants	mean height	variance
CentralAmerican	773	160	73.7
SouthAmerican	499	160	81.6
Mexican	2688	161	86.2
PuertoRican	1298	163	89.3
Cuban	1234	164	89.5
Dominican	670	162	78.9
All combined	7162	162	87.2

Dealing with heterogeneity (1) heterogeneous variances

Recall the linear mixed model for quantitative traits:

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

- with b_i a random error - or a random effect - in addition to the i.i.d. error ϵ_i.

The usual assumption is that $\text{var}(\epsilon_i) = \sigma_e^2$ for all $i = 1, \ldots, n$.

- If there are well-defined sub-groups, one can fit a model which assigns each group its own residual variance.
  - Heterogeneous variances.

Dealing with heterogeneity (1) heterogeneous variances

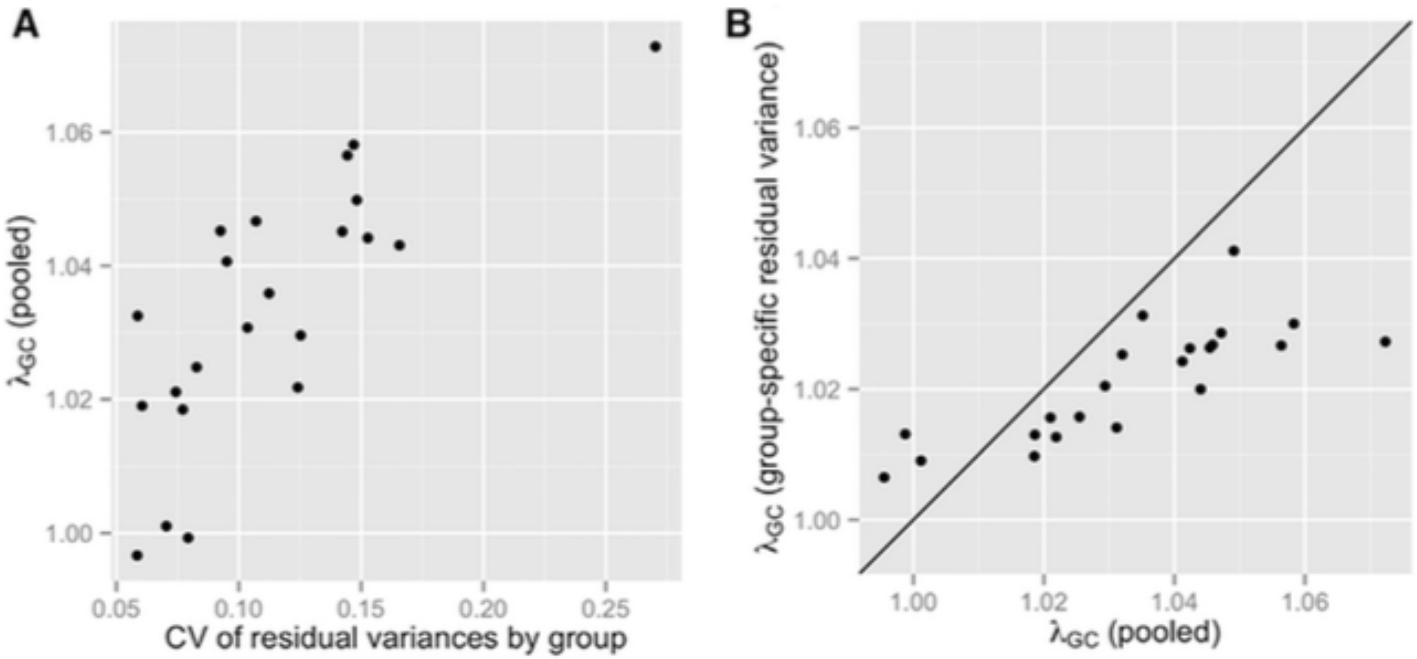
Suppose that each participant i is associated with a subgroup $k = 1, \ldots, K$. Then:

$$y_{ik} = \mathbf{x}_{ik}^T \beta + g_{ik}\alpha + b_{ik} + \epsilon_{ik}, i = 1, \ldots, n, k = 1, \ldots, K$$

- with subscript k added to denote that person i is in group k.
- With homogeneous variances: $\text{Var}(\epsilon_{ik}) = \sigma_e^2$ for all $i = 1, \ldots, n$. (The usual model)
- With heterogeneous variances: $\text{Var}(\epsilon_{ik}) = \sigma_{e,k}^2$.

Dealing with heterogeneity (1) heterogeneous variances

In the HCHS/SOL, we showed that using heterogeneous variances is useful for control of inflation, in GWASs of 22 traits.



From: Conomos et al. 2016, Genetic Diversity and Association Studies in US Hispanic/Latino Populations: Applications in the Hispanic Community Health Study/Study of Latinos, AJHG.

Linear mixed models with heterogeneous variances

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 with heterogeneous variances

```
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 <- "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 with heterogeneous variances

Note the group.var argument!

```
nullmod <- fitNullMM(scanData = scanAnnot,
                     outcome = outcome, covars = covariates,
                     covMatList = covMatList,
                     group.var = "group", verbose = FALSE)
```

Linear mixed models with heterogeneous variances

- 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_uw     V_unc
##   25.87177    0.00000 455.99390  42.18352
```

Linear mixed models with heterogeneous variances

- Let's look at the results:

```
nullmod$fixef
```

```
##                    Est          SE          Stat           pv
## (Intercept)   3.137328 1.59805848      3.854202 4.962152e-
## EV1           5.126688 0.46027506    124.062015 8.165140e-
## EV2          -3.682425 0.49397580     55.572048 9.009564e-
## sexM          3.990980 0.91490367     19.028664 1.287695e-
## age           3.822844 0.03297316  13441.638824 0.000000e-
## groupuw      -4.912214 1.62688357      9.116784 2.532748e-
```

- Unfortunately, function varCompCI(nullmod, prop = TRUE) is not supported for heterogeneous variances.
  - It does not calculate heritability for each group separately. Maybe it should?

Association testing proceeds as usual.

```
gds <- GdsGenotypeReader(file.path(dir,
                                   "SISG_snp_dosages.gds"))
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 - 1.551 secs

## Block 2 of 2 Completed - 0.6163 secs
```

Association testing proceeds as usueal.

```
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  8.6735196 8.3272966
## 3     3   1 500 0.008            A -2.2725077 3.5415649
## 4     4   1 500 0.000            A         NA         NA
## 5     5   1 500 0.209            B -0.1156652 0.7519208
## 6     6   1 500 0.174            B -1.6718879 0.8278857
##    Wald.pval
## 1         NA
## 2 0.29760789
## 3 0.52108900
## 4         NA
## 5 0.87774666
## 6 0.04343868
```

```
close(gds)
```

Exercises

1. Compare the p-values and effect estimates using the plot() command from GWAS with and without heterogeneous variances.
   - You can compare -log(p-value,10).
2. Can you use heterogeneous variances in logistic regression?

Dealing with heterogeneity (2) stratified analysis

- Heterogeneous variance model accounts for differences in residual variances.
- While other effects remain common to all groups (e.g. fixed effects, genetic variances).
- Stratified analysis allows for a different model, with different parameters, in each stratum.
- One can inspect results in each stratum, and combine results across strata in meta-analysis.

Dealing with heterogeneity (2) stratified analysis

- Combining results across strata in meta-analysis is not trivial, due to genetic relatedness and shared household (environmental correlation) among individuals in different groups.
- Method to do it: MetaCor. Meta-analysis when correlations between strata exist, and are modeled.
  - Mixed-models based approach.

Sofer et al. (2016), "Meta-Analysis of Genome-Wide Association Studies with Correlated Individuals: Application to the Hispanic Community Health Study/Study of Latinos (HCHS/SOL)", gen epi.

Dealing with heterogeneity (2) stratified analysis

It may help with inflation control. Example: log(BMI) analysis.

Compare stratifications (complete data set)

Stratification	method	λ_{GC}
None	Pooled	1.050
sex	MetaCor	1.048
Ethnic	MetaCor	1.034
Sex + ethnic	MetaCor	1.028

Stratification reduces inflation!

Compare tests (stratification by sex and ethnic background group)

Dataset	method	λ_{GC}
Complete	MetaNaive	1.088
Distant	StratInd	1.058
Distant	MetaCor	1.027

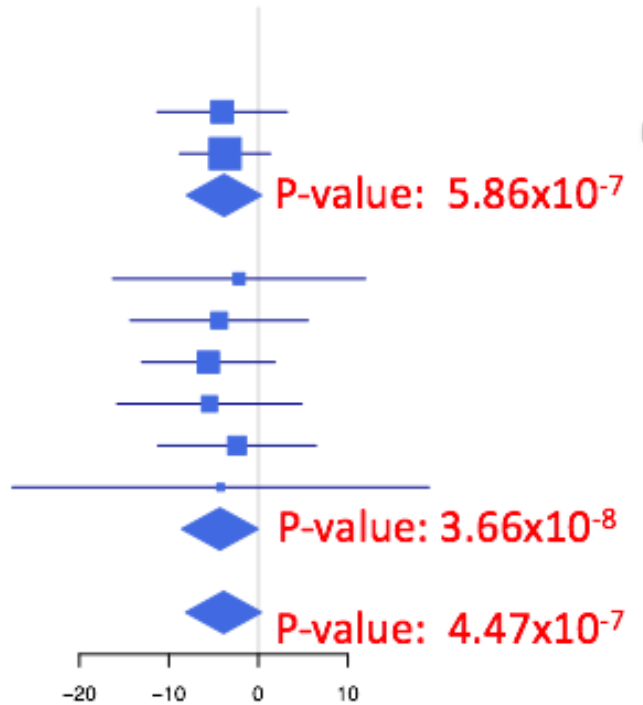
Ignoring correlations between strata = bad!

Accounting for even small correlations, due to distant relatives = good!

Dealing with heterogeneity (2) stratified analysis

It may help with power. Example: significant variant detected in dental caries (tooth decays) analysis.

Analysis	EAfreq	Effect size	1−(5e−08)% Confidence Interval
ever	0.969	−4.043	(−11.288 , 3.201)
never	0.969	−3.722	(−8.783 , 1.339)
meta_smoke		**−3.827**	**(−8.004 , 0.349)**
CentralAmerican	0.976	−2.153	(−16.262 , 11.956)
Cuban	0.962	−4.374	(−14.3 , 5.552)
Dominican	0.889	−5.58	(−13.018 , 1.858)
Mexican	0.989	−5.463	(−15.783 , 4.857)
PuertoRican	0.959	−2.383	(−11.225 , 6.458)
SouthAmerican	0.985	−4.211	(−27.497 , 19.074)
meta_ethnic_group		**−4.288**	**(−8.539 , −0.038)**
pooled	**0.967**	**−3.884**	**(−8.086 , 0.318)**



Dealing with heterogeneity (2) stratified analysis

We can use the MetaCor package. If not already installed, install using:

```
library(devtools)
install_github("tamartsi/MetaCor")
```

Tell R we want to use it:

```
require(MetaCor)
```

Dealing with heterogeneity (2) stratified analysis

. . . load the data again.

```
gds <- GdsGenotypeReader(file.path(dir,
                                   "SISG_snp_dosages.gds"))
snpAnnot <- getobj(file.path(dir,
                             "SISG_snp_dosages_snpAnnot.RData"))
genoData <- GenotypeData(gds,
                  snpAnnot=snpAnnot, scanAnnot = scanAnnot)
```

Dealing with heterogeneity (2) stratified analysis

- Using the MetaCor package is slightly more complicated then using GENESIS.
- It takes as arguments actual genotype counts and covariates info.
- So we need to extract them first.
- We can use a function from the GENESIS package.

```
covariates <- c("EV1", "EV2", "sex", "age")
 dat <- GENESIS:::createDesignMatrix(
            scanData = pData(scanAnnot),
              outcome = outcome,
              covars = covariates,
              scan.include = scanAnnot$scanID)
```

Dealing with heterogeneity (2) stratified analysis

- We got an object that has the outcome, and a design matrix.
- Note that we do not use the "group" variable as a covariate, because we are going to use it for stratification!

```
names(dat)
```

```
## [1] "Y" "W" "k"
```

```
dat$k
```

```
## [1] 5
```

```
# k is just the number of colums in the design matrix W.
```

```
head(dat$Y)
```

```
## [1]  30.54507 243.39091 203.80642 217.49060 148.45333 17
```

Dealing with heterogeneity (2) stratified analysis

```
head(dat$W)
```

```
##    (Intercept)         EV1         EV2 sexM age
## p1           1  0.8199808  0.0845245    0  20
## p2           1  3.1232047  1.9885938    0  60
## p3           1  0.5849367 -1.3709960    0  42
## p4           1 -0.5743757  2.0681971    1  60
## p5           1 -2.0852808  0.1698396    0  41
## p6           1 -0.7885409 -0.5235238    1  45
```

Dealing with heterogeneity (2) stratified analysis

- Let's set the names of the outcome Y to be the sample IDs.
- The correlation matrix kin.mat are already in our workspace, but let's load a different household matrix.
  - And their column and row names are the sample IDs.

```
W <- dat$W
Y <- dat$Y
names(Y) <- rownames(dat$W)
HH.mat <- getobj(file.path(dir,
                           "SISG_houshold_matrix_2.RData"))
HH.mat[1:3,1:3]
```

```
##    p1 p2 p3
## p1  1  0  0
## p2  0  1  0
## p3  0  0  1
```

Dealing with heterogeneity (2) stratified analysis

- To stratify by group, we now need to prepare a list with sample IDs for each group.

```
covMatList <- list(HH = HH.mat, kinship = kin.mat)
IDs.list.group <- list(
  uw =  scanAnnot$scanID[which(scanAnnot$group == "uw")],
  unc = scanAnnot$scanID[which(scanAnnot$group == "unc")])
```

Dealing with heterogeneity (2) stratified analysis

- Finally, let's read the genotype from file.

```
G <- getGenotype(genoData)
dim(G)
```

```
## [1] 7463  500
```

```
G[3:7,5:7]
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    1    0
## [4,]    1    1    0
## [5,]    1    1    0
```

Dealing with heterogeneity (2) stratified analysis

- Assign sample and SNP IDs to the rows and columns of G:
- And make the rows of G correspond to people by transposing...

```
colnames(G) <- getScanID(genoData)
rownames(G) <- getSnpID(genoData)
G[3:7,5:7]
```

```
##   p5 p6 p7
## 3  0  0  0
## 4  0  0  0
## 5  0  1  0
## 6  1  1  0
## 7  1  1  0
```

```
G <- t(G)
```

Dealing with heterogeneity (2) stratified analysis

- Finally we can perform stratified analysis!

```
strat.model <- stratLMMTest(Y = Y, W = as.matrix(W),
                            G = as.matrix(G[,c(1:100)]),
                            covMatList = covMatList,
                            IDsList = IDs.list.group,
                            verbose = FALSE,
                            testType = "MetaGLS")
```

```
## Some missing outcome, covariates and/or genotypes. Remov

## Strata are assumed independent according to covariance h
```

Dealing with heterogeneity (2) stratified analysis

- I used only first 100 SNPs, because there are so many missing SNPs when considering all variants, too many people were removed and the algorithm did not converge.
- Alternatively, I could have imputed allele counts to the mean.

```
close(gds)
```

Exercises

1. Run the stratified analysis - for all variants. This can be done in groups of 100 variants at a time, say, with results merged later.
   - Initiate a list to hold results using `res.list <- vector(mode = "list", length = ceiling(ncol(G)/100))`
   - Assign results to list slots using `res.list[[i]] <- strat.model`.
   - Merge results using `assoc <- do.call(rbind, res.list)`.
2. Calculate p-values for the group-specific results using p `<- pchisq(beta^2/var(beta), df = 1, lower.tail = FALSE)`.
   - Which are the SNPs with the most significant p-value in each of the analyses? what are their p-value in the other analyses?
3. Plot the UW-specific, UNC-specific, and meta-analysis p-values against each other (all combinations).
4. Fit sex-stratified analysis instead of group-stratified.
5. How would you stratify by BOTH sex AND UW/UNC group indicators?

Exercises

6. Different softwares handle missing data differently by default. MetaCor removes people with missing genotypes in the group of tested variants. In this toy data sets there are many individuals with missing values. Use the following code to impute SNP dosages to the mean, and re-run stratLMMTest(). You can supply all variants to the function, as in `G = as.matrix(G)`.

```
for (i in 1:ncol(G)){
  inds.mis <- which(is.na(G[,i]))
  G[inds.mis,i] <- mean(G[,i], na.rm = TRUE)
}
```