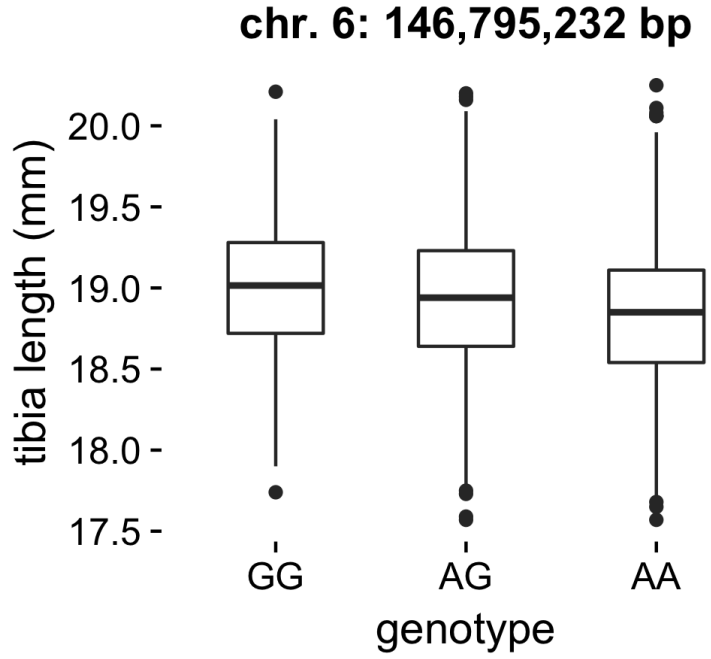
Analysis of Genetic Data 2: Mapping Genome-Wide Associations

Peter Carbonetto

Research Computing Center and the Dept. of Human Genetics
University of Chicago



Workshop aims

1. Implement basic steps of a genome-wide association analysis.
2. Understand how phenotype and genotype data for a genome-wide association study (GWAS) are encoded in computer files.
3. Understand some of the benefits (and complications) of using linear-mixed models (LMMs) for GWAS.
4. Use command-line tools to inspect and manipulate phenotype and genotype data.

Workshop aims

- This is a *hands-on workshop*—you will get the most out of this workshop if you work through the exercises on your computer.
- The examples are intended to run either on the RCC cluster or on a laptop with Linux or macOS (not Windows).
- About 15 GB will be needed to store all the the data and output files.
- *Note: the examples may not produce the exact same result on your laptop. Using the RCC cluster is recommended.*

Software we will use today

1. GEMMA
2. R
3. Several R packages: data.table, ggplot2, cowplot and stringr.
4. Basic shell commands such as “cat” and “wc”.

Our research task

Identify and interpret genetic loci contributing to tibia bone development in mice.

1. We will use data from this study:
   - ▷ Nicod *et al* (2016). “Genome-wide association of multiple complex traits in outbred mice by ultra-low-coverage sequencing.” *Nature Genetics* **48:** 912–918.
2. To map tibia development genes, we will assess support (*p*-values) for statistical association between tibia length and 353,697 SNP genotypes.
3. We will visualize the association signal to identify candidate genes for tibia bone development.

Background on study

- Samples are from an outbred mouse population, “Carworth Farms White” (CFW).
- 200 measurements (blood concentrations, muscle weights, *etc*) were taken for each mouse.
- Genotypes were obtained from low-coverage DNA sequencing.
- Size of data is comparable to human GWAS: 2,000 samples, 350,000 SNPs.
- Linkage disequilibrium (LD) does not decay as rapidly as humans, so mapping resolution will not be as good as a human GWAS.
- Data are publicly available.

Outline of workshop

- **Preliminaries**
- Programming challenges:
  1. Setting up your environment for an association analysis.
  2. Preparing the data for an association analysis.
  3. Running association analyses in GEMMA.
  4. Visualizing and interpreting the results of an association analysis.

Preliminaries

- WiFi.
- Power outlets.
- Reading what I type.
- Pace, questions (e.g., keyboard shortcuts).
- YubiKeys.
- What to do if you get stuck.

Preliminaries

- The workshop packet is a repository on GitHub. Go to:
  - ▷ github.com/rcc-uchicago/genetic-data-analysis-2
- Download the workshop packet to your computer.
- If necessary, unzip the ZIP file.

What's included in the workshop packet

- **slides.pdf:** These slides.
- **slides.Rmd:** R Markdown source used to create these slides.
- **download_genotypes.sh:** Script for downloading the genotype data.
- **functions.R**: Some R functions for loading the data and creating plots.
- **format.genotypes.for.gemma.R:** Script to convert genotype data into a format suitable for GEMMA.
- **CFW_measures.txt:** a large table with 200 columns (phenotypes) and 2,117 rows (samples).
- **CFW_covariates.txt:** covariate data for 2,117 CFW mice.
- **listof1934miceusedforanalysis.txt:** ids of the 1,934 mice used in the analyses of Nicod *et al* (2016).

Outline of workshop

- Programming challenges:
  1. **Setting up your environment for an association analysis.**
  2. Preparing the data for an association analysis.
  3. Running association analyses in GEMMA.
  4. Visualizing and interpreting the results of an association analysis.

Challenge #1: Setting up your environment for GWAS

- Aim: Configure your computing environment for the next programming challenges.
- Steps:
  1. Connect to midway2.*
  2. Clone git repository.*
  3. Download genotype data.
  4. Download GEMMA.
  5. Connect to a midway2 compute node.*
  6. Launch R, and check your R environment.
  7. Set up R for plotting.

Connect to midway2

- **If you have an RCC account:** I'm assuming you already know how to connect to midway2. *ThinLinc is recommended if you do not know how activate X11 forwarding in SSH.*
  - ▷ See: rcc.uchicago.edu/docs/connecting
- **If you do not have an RCC account:** I can provide you with a Yubikey. This will give you guest access to the RCC cluster (see the next slide).

Using the Yubikeys

- Prerequisites:
  1. SSH client (for Windows, please use **MobaXterm**)
  2. USB-A port
- Steps:
  1. Insert Yubikey into USB port.
  2. Note your userid: `rccguestXXXX`, where `XXXX` is the last four digits shown on Yubikey.
  3. Follow instructions to connect to midway2 via SSH, replacing the cnetid with your `rccguestXXXX` user name: rcc.uchicago.edu/docs/connecting
  4. When prompted for password, press lightly on metal disc.
- Important notes:
  - ▷ Yubikeys do not work with ThinLinc.
  - ▷ *Please return the Yubikey at the end of the workshop.*

Clone git repository

Clone the workshop packet in your home directory on midway2 (**note:** there are no spaces in the URL below):

```
cd $HOME
git clone https://github.com/rcc-uchicago/
  genetic-data-analysis-2.git
cd genetic-data-analysis-2
```

Alternatively, it may be simpler to download and unzip the file containing the contents of the git repository:

```
cd $HOME
wget -O repo.zip https://tinyurl.com/2p9caa2m
unzip repo.zip
mv genetic-data-analysis-2-master \
  genetic-data-analysis-2
```

Download genotype data

The genotype data are too large to fit in the git repository. So we need to download these data separately.

- Go to: `http://mtweb.cs.ucl.ac.uk/dosages`
- Download files for chromosomes 1–19.
- You may find it easier to run the provided bash script:

  ```
  bash download_genotypes.sh
  ```
- After following these steps, you should have a total of 19 files with extension ".RData" in the "genetic-data-analysis-2" folder.

Alternatively, download the prepared data (faster):

```
wget users.rcc.uchicago.edu/~pcarbo/cfwgeno.tar.gz
tar zxvf cfwgeno.tar.gz
```

Download GEMMA

Download GEMMA 0.96. Go to:

- `https://github.com/genetics-statistics/GEMMA/releases/tag/v0.96`

On midway2, you can run these commands to download and install GEMMA 0.96 for Linux:

```
wget http://bit.ly/2I67fBV -O gemma.linux.gz
gunzip gemma.linux.gz
mv gemma.linux gemma
chmod 700 gemma
./gemma -h
```

Connect to a midway2 compute node

Set up an interactive session on a midway2 compute node with 4 CPUs and 10 GB of memory:

```
screen -S workshop
sinteractive --partition=broadwl \
  --reservation=gda2_workshop --cpus-per-task=4 \
  --mem=10G --time=3:00:00 --account=pi-bigcheese
echo $HOSTNAME
```

Launch R

Start up an interactive R session. On midway2, you can start up R with these commands:

```
module load R/4.2.0
which R
R --no-save
```

Note: You can use R or RStudio.

Check your R environment

Check the version of R you are running:

```
sessionInfo()
```

Check that you are starting with an empty environment:

```
ls()
```

Check that you have the correct working directory—it should be set to the “genetic-data-analysis-2” repository:

```
getwd()
```

Set up R for plotting

Make sure you can display graphics in your current R session.

```
library(ggplot2)
data(cars)
qplot(cars$dist,cars$speed)
```

You should see a scatterplot. If not, your connection is not set up to display graphics. On midway2, an alternative is to save the plot to a file, and download the file to your computer using sftp or SAMBA. See:

- rcc.uchicago.edu/docs/data-transfer

Another alternative is to run R or RStudio on your computer, and set your working directory to the genetic-data-analysis-2 directory mounted via SAMBA.

Quit R

We will quit R, and return to it later.

```
quit()
```

Outline of workshop

- Preliminaries
- Programming challenges:
  1. Setting up your environment for an association analysis.
  2. **Preparing the data for an association analysis.**
  3. Running association analyses in GEMMA.
  4. Visualizing and interpreting the results of an association analysis.

Challenge #2: Prepare the data for GWAS

- Aim: Perform an exploratory analysis of the phenotype and covariate data, and prepare the data for association analysis with GEMMA.
- Steps:
  1. Prepare genotype data for GEMMA.
  2. Import phenotype and covariate data into R.
  3. Explore phenotype and covariate data in R.
  4. Save phenotype and covariate data for GEMMA.

Prepare the genotype data for GEMMA

Processing the genotype data is more complicated, so I have provided an R script to create, for each chromosome, a SNP annotation file and a genotype file in the format used by GEMMA. Run the script in R:

```
Rscript format.genotypes.for.gemma.R
```

This will take a few minutes to run.

- Note that you can skip this step if you downloaded the `cfwgeno.tar.gz` file.

Prepare the genotype data for GEMMA

The genotype data need to be combined into one file. This can be easily done with the `cat` command, which combines by lines (rows):

```
cat chr*.geno.txt > geno.txt
```

We do the same for the SNP annotations:

```
cat chr*.map.txt > map.txt
```

- Again, you can skip this step if you downloaded the `cfwgeno.tar.gz` file.

Import phenotypes and covariates into R

Start up R. On midway2, run:

```
R --no-save
```

Before continuing, verify that your R working directory is set to the "genetic-data-analysis-2" repository:

```
getwd()
```

Import phenotypes and covariates into R

Load the phenotype and covariate data:

```
source("functions.R")
dat1 <- read.pheno("CFW_measures.txt")
dat2 <- read.pheno("CFW_covariates.txt")
```

Merge the two data frames, dropping the redundant “id” column:

```
pheno <- cbind(dat1,dat2[-1])
```

Select samples

We will use the same 1,934 samples that were used in the published association analysis (Nicod *et al*, 2016). First, load the selected sample ids:

```
ids <-
  read.table("listof1934miceusedforanalysis.txt",
             stringsAsFactors = FALSE)
ids <- ids[[1]]
```

Next, select the rows of the phenotype table that match up with the ids:

```
rows  <- match(ids,pheno$Animal_ID)
pheno <- pheno[rows,]
```

Examine tibia length

Take a closer look at the data for the phenotype of interest, tibia length (units are “mm”):

```
summary(pheno$Tibia.Length)
```

Plot the distribution of tibia length:

```
library(ggplot2)
library(cowplot)
theme_set(theme_cowplot())
p1 <- ggplot(pheno,aes(Tibia.Length)) +
  geom_histogram(fill = "darkblue",
                 na.rm = TRUE)
print(p1)
```

Examine tibia length vs. body weight

The study authors recommend using body weight as a covariate in the association analysis (units are “g”). What do the data tell us about the relationship between the two?

```
summary(pheno$Weight.Diss)
```

Visualize the relationship between tibia length and body weight:

```
p2 <- qplot(pheno$Weight.Diss,
            pheno$Tibia.Length,
            na.rm = TRUE)
print(p2)
```

Examine tibia length vs. sex

The study authors also recommend using sex as a covariate in the association analysis. Let's examine the data:

```
pheno <- transform(pheno,Sex = factor(Sex))
summary(pheno$Sex)
```

A boxplot is a common way to visualize the relationship between a continuous variable (tibia length) and a categorical variable (sex):

```
p3 <- ggplot(pheno,aes(Sex,Tibia.Length)) +
      geom_boxplot(na.rm = TRUE)
print(p3)
```

Verify phenotype residuals

Compute the phenotypes after removing the linear effects of sex and body weight:

```
fit <- lm(Tibia.Length ~ Sex + Weight.Diss,pheno)
r   <- resid(fit)
```

Plot the distribution of the residuals:

```
p4 <- ggplot(data.frame(resid = r),aes(resid)) +
      geom_histogram(fill = "darkblue")
print(p4)
```

Save phenotype data for GEMMA

Now that we have carefully examined the phenotype data, we are ready to save the data in a format usable by GEMMA. First write the tibia measurements to a new file, “pheno.txt”:

```
write.table(pheno$Tibia.Length,"pheno.txt",
            row.names = FALSE,
            col.names = FALSE)
```

Save covariate data for GEMMA

Next, write the covariates (body weight and sex) to new file, “covar.txt”, in the format used by GEMMA. *Sex needs to be encoded as a number:*

```
sex <- as.numeric(pheno$Sex) - 1
dat <- data.frame(1,sex,pheno$Weight.Diss)
write.table(dat,"covar.txt",sep = " ",
            row.names = FALSE,
            col.names = FALSE)
```

Preparing the phenotype data: take-home points

Most analyses will involve:

- Carefully checking phenotype and covariate data for possible issues, and resolving these issues (e.g., filtering out problematic samples, transforming phenotypes).
- Reformatting the data.

Here we performed these steps in R.

Preparing the genotype data: take-home points

- Typically, this is much more work (fortunately, the authors provided the processed genotype data).
- A common mistake is to save the phenotype and genotype samples in different orders, which will lead to an incorrect association analysis!

Outline of workshop

- Preliminaries
- Programming challenges:
  1. Setting up your environment for an association analysis.
  2. Preparing the data for an association analysis.
  3. **Running association analyses in GEMMA.**
  4. Visualizing and interpreting the results of an association analysis.

Challenge #3: Run association analyses

- Aim: Compute *p*-values to assess support for association between tibia length and SNPs on chromosomes 1–19.
- Steps:
  1. Run a basic association analysis in GEMMA.
  2. Run an LMM-based association analysis in GEMMA.
  3. Compare the two association analyses.

Basic association analysis in GEMMA

Now that we have installed GEMMA, and we have prepared the phenotype and genotype data in the formats accepted by GEMMA, we are now ready to run our first association analysis.

```
./gemma -p pheno.txt -c covar.txt \
  -a map.txt -g geno.txt -notsnp \
  -lm 2 -outdir . -o tibia
```

This should take no more than a few minutes to complete.
To view the results of the association alaysis, run

```
column -t -o ' ' tibia.assoc.txt | less -S
```

Basic association analysis in GEMMA

This GEMMA call creates two files:

- **tibia.log.txt:** A summary of the association analysis.
- **tibia.assoc.txt:** The full table of association results (*p*-values, effect size estimates, *etc*).

LMM-based association analysis in GEMMA

First, compute a kinship ("realized relatedness") matrix from the genotypes.

```
./gemma -p pheno.txt -g geno.txt \
  -a map.txt -gk 1 -outdir . -o tibia
```

Next, compute LMM-based association statistics for each SNP:

```
./gemma -p pheno.txt -c covar.txt \
  -a map.txt -g geno.txt -notsnp \
  -k tibia.cXX.txt -lmm 2 -outdir . \
  -o tibia_lmm
```

LMM-based association analysis in GEMMA

Since the LMM-based association analysis may take a while to complete (perhaps an hour or more), I have provided files containing the outputs of these two GEMMA commands:

- **tibia.cXX.txt.gz:** The 1,934 x 1,934 kinship matrix.
- **tibia_lmm.log.txt:** A summary of the association analysis.
- **tibia_lmm.assoc.txt.gz:** The full table of association results.

To use these files, you need to uncompress them first:

```
gunzip tibia.assoc.txt.gz
gunzip tibia_lmm.assoc.txt.gz
```

Compare basic and LMM association analyses

The LMM is expected to reduce inflation of small *p*-values; a high level of inflation could indicate many false positive associations.

- The q-q plot is commonly used to assess inflation. This test is useful as a simple, heuristic diagnostic.

Launch R again, and import the results of the first GEMMA association analysis:

```
gwscan1 <- read.table("tibia.assoc.txt",
                      as.is = "rs",header = TRUE)
```

Compare basic and LMM association analyses

Plot the observed *p*-values against the expected *p*-values under the null distribution:

```
library(ggplot2)
library(cowplot)
theme_set(theme_cowplot())
source("functions.R")
p1 <- plot.inflation(gwscan1$p_lrt)
print(p1)
```

Next, we will compare this plot against the LMM q-q plot.

Compare basic and LMM association analyses

Follow the same steps as before to load the GEMMA results and create a genomic inflation plot:

```
gwscan2 <- read.table("tibia_lmm.assoc.txt",
                      as.is = "rs",header = TRUE)
p2 <- plot.inflation(gwscan2$p_lrt)
print(p2)
```

Compare against the previous inflation plot.

LMM-based association analysis: take-home points

- LMMs are now one of the most widely used approaches to reduce inflation of false positive associations due to population structure and hidden relatedness.
- LMMs can be computationally intensive, especially for data sets with many samples.
- LMMs can *overcorrect* inflation. One remedy is the "leave-one-chromosome-out" (LOCO) approach.

Outline of workshop

- Preliminaries
- Programming challenges:
  1. Setting up your environment for an association analysis.
  2. Preparing the data for an association analysis.
  3. Running association analyses in GEMMA.
  4. **Visualizing and interpreting the results of an association analysis.**

Challenge #4: Visualize and interpret results of association analysis

- Aim: Create plots to gain insight into the results of the association analysis.
- Steps:
  1. Summarize “genome-wide” association signal for tibia length.
  2. Visualize strongest association signal.
  3. Based on the association signal, identify candidate genes.
  4. Visualize the relationship between tibia length and the top SNP association.

Plot genome-wide scan for tibia length

Using GEMMA, we computed association *p*-values at 353,697 SNPs on chromosomes 1–19. Now let's plot these *p*-values to get a visual summary of the association results. First, load the results of the LMM association analysis:

```
gwscan <- read.table("tibia_lmm.assoc.txt",
                     as.is = "rs",header = TRUE)
```

Plot genome-wide scan for tibia length

I wrote a function that uses `ggplot2` to create a "Manhattan plot" showing the results of the association analysis.

```
library(ggplot2)
library(cowplot)
theme_set(theme_cowplot())
source("functions.R")
p1 <- plot.gwscan(gwscan)
print(p1)
```

This gives us a "genome-wide" view of the association signal.

Examine the association signal on chromosome 6

Here we look more closely at the strongest association signal on chromosome 6. First, create a data frame with the results for chromosome 6 only:

```
gwscan_chr6 <- subset(gwscan,chr == 6)
```

Next, plot the *p*-values against the base-pair position of the SNPs.

```
p2 <- plot.region.pvalues(gwscan_chr6)
print(p2)
```

Examine the association signal on chromosome 6

It appears that the strongest association signal is isolated to positions of 145 Mb and greater. Let's zoom in on that region:

```
dat <- subset(gwscan_chr6,ps > 145e6)
p3 <- plot.region.pvalues(dat)
print(p3)
```

Let's find the SNP on chromosome 6 that is most strongly associated with tibia length:

```
i <- which.min(gwscan_chr6$p_lrt)
gwscan_chr6[i,]
```

Visualize genotype-phenotype relationship

Above, we identified the SNP that is most strongly associated with tibia length.

- Next, we examine the relationship between tibia length and the genotype at this SNP.

Load the GEMMA phenotype data:

```
pheno <- read.table("pheno.txt")
pheno <- pheno[[1]]
```

Visualize genotype-phenotype relationship

Load the GEMMA genotype data for chromosome 6, and retrieve the genotypes for the top SNP. To obtain discrete genotype values (0, 1, 2), round the numeric values to the nearest integer.

```
command <- paste("grep snp-6-146795232 geno.txt >",
                 "geno_chr6_146795232.txt")
system(command)
geno <- read.table("geno_chr6_146795232.txt",
                   sep = " ")
geno <- as.numeric(geno)
geno <- geno[-(1:3)]
```

Visualize genotype-phenotype relationship

If we convert the genotype to a “factor”, we can visualize the relationship with a boxplot.

```
x <- round(geno)
x <- factor(x,0:2)
levels(x) <- c("GG","AG","AA")
```

A is the alternative allele, so a 2 corresponds to genotype “AA”.

```
pdat <- data.frame(pheno = pheno,x = x)
p <- ggplot(pdat,aes(x,pheno)) +
     geom_boxplot(na.rm = TRUE)
print(p)
```

This provides a visual summary of the genotype-phenotype relationship. We can also quantify the relationship:

```
fit <- lm(y ~ x,data.frame(x = geno,
                           y = pheno))
summary(fit)
```

Recap

- We implemented the basic steps of a GWAS using GEMMA, R and command-line tools such as “cat”.
- We saw that much of the effort went into data processing and quality control.
- We did not discuss how to determine an appropriate significance threshold. This is a complicated question. *My advice:* first take an exploratory approach to identify the strongest association signals, and determine significance later.