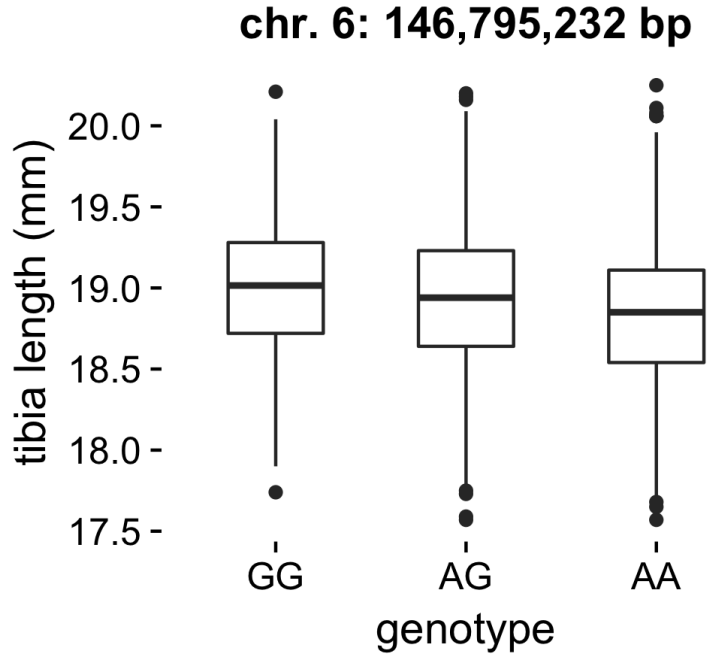
Analysis of Genetic Data 2: Mapping Genome-Wide Associations

Peter Carbonetto

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



Workshop aims

1. Implement the 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)
- *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: stringr, data.table, ggplot2 and cowplot.
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, unlike most human studies.

Outline of workshop

- **Preliminaries**
- Programming challenges:
  1. Setting up your environment for GWAS.
  2. Preparing the data for GWAS.
  3. Running a basic association analysis in GEMMA.
  4. Running an LMM-based association analysis in GEMMA.
  5. Visualizing and interpreting the results of the 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, rename the folder to “genetic-data-analysis-2”.

What's included in the workshop packet

- **slides.pdf:** These slides.
- **slides.Rmd:** R Markdown source used to create these slides.
- **CFW_measures.txt:** a large table with 200 columns (phenotypes) and 2,117 rows (sampels).
- **CFW_covariates.txt:** covariate data for 2,117 CFW mice.
- **listof1934miceusedforanalysis.txt:** ids of the 1,934 mice used in the final analyses of Nicod *et al* (2016).

Outline of workshop

- Programming challenges:
  1. **Setting up your environment for GWAS.**
  2. Preparing the data for GWAS.
  3. Running a basic association analysis in GEMMA.
  4. Running an LMM-based association analysis in GEMMA.
  5. Visualizing and interpreting the results of the 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
```

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
  ```

  This script requires “wget”.
- After following these steps, you should have a total of 19 files with extension “.RData” in the “genetic-data-analysis-2” folder.

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=workshop --cpus-per-task=4 \
  --mem=10G --time=3:00:00
echo $HOSTNAME
```

Launch R

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

```
module load R/3.5.1
which R
R --no-save
```

Note: You may use RStudio instead of R.

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)
library(cowplot)
data(cars)
quickplot(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

Quit R

We will quit R, and return to it later.

```
quit()
```

Outline of workshop

- Preliminaries
- Programming challenges:
  1. Setting up your environment for GWAS.
  2. **Preparing the data for GWAS.**
  3. Running a basic association analysis in GEMMA.
  4. Running an LMM-based association analysis in GEMMA.
  5. Visualizing and interpreting the results of the 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. Import phenotype and covariate data into R.
  2. Explore phenotype and covariate data in R.
  3. Save phenotype and covariate data for GEMMA.
  4. Prepare genotype data for GEMMA.

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)[[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 main phenotype of interest —tibia length (units are “mm”):

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

Plot the distribution of tibia length:

```
library(ggplot2)
library(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?

```
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 the most common w ay 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(color = "darkblue")
```

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

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:

```
source("format.genotypes.for.gemma.R")
```

This will take a few minutes to run. Once it is done, quit R.

Prepare the genotype data for GEMMA (part 2)

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
head geno.txt | less -S
tail geno.txt | less -S
wc -l geno.txt
```

We do the same for the SNP annotations:

```
cat chr*.map.txt > map.txt
wc -l map.txt
```

Why didn't we combine the genotypes inside the R script?

Preparing the phenotype data: take-home points

Most association analyses will involve these steps:

1. Carefully checking phenotype and covariate data for possible issues, and resolving these issues (e.g., filtering out problematic data, transforming phenotypes).
2. Aligning phenotypes to genotypes.
3. Reformatting the data.

In this tutorial, we performed these steps in R.

Preparing the genotype data: take-home points

- Typically this is much more work—*we were fortunate that the authors provided the post-processing, post-QC (quality control) genotype data.*
- A **critical step** in this script was to make sure that the samples in the genotype and phenotype files are listed in the same order. *A common mistake is to save the phenotype and genotype samples in different orders, which will lead to an incorrect association analysis!*
- Why is the PLINK file format helpful for preventing such errors?

Outline of workshop

1. Initial setup.
2. Download genotype data.
3. Prepare phenotype data.
4. Prepare genotype data.
5. **Run a basic association analysis in GEMMA, and assess quality of results.**
6. Run an LMM-based association analysis in GEMMA, and compare against the initial analysis.
7. Visualize the genome-wide association signal in R to (a) map tibia loci and (b) identify plausible tibia genes.

Basic association analysis in GEMMA (part 1)

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.

```
cd data
../bin/gemma -p pheno.txt -c covariates.txt \
  -a map.txt -g geno.txt -notsnp -lm 2 \
  -outdir ../output -o tibia
```

This should take no more than a few minutes to complete.

- The `-notsnp` option is included here to prevent any automatic processing of the genotype data.

Basic association analysis in GEMMA (part 2)

This GEMMA command creates two files:

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

Use `less -S`, or your favourite text editor, to inspect these files.

Outline of workshop

1. Initial setup.
2. Download genotype data.
3. Prepare phenotype data.
4. Prepare genotype data.
5. Run a basic association analysis in GEMMA, and assess quality of results.
6. **Run an LMM-based association analysis in GEMMA, and compare against the initial analysis.**
7. Visualize the genome-wide association signal in R to (a) map tibia loci and (b) identify plausible tibia genes.

LMM association analysis in GEMMA

First, generate a "realized relatedness" matrix from the (centered) genotypes. This may take 10–15 minutes.

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

Next, fit an LMM separately for each SNP:

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

These two steps may take an hour or longer combined, *so I have provided a compressed file in the* `output` *directory containing the result of this command. To use this file, run

```
gunzip tibia-lmm.assoc.txt.gz
```

Optional: Compare "genomic inflation" (part 1)

The LMM is expected to reduce inflation of small *p*-values; a high level of inflation could indicate many *false positive* associations (e.g., due to populations tructure or "hidden" relatedness). *Here we will check that the LMM does this.*

- The q-q plot is commonly used to assess inflation. This test has many well-known problems, but is nonetheless useful as a simple diagnostic.

Start up R again, and use `read.table` to load the results of the GEMMA association analysis:

```
getwd() # Should be .../code
gwscan <- read.table("../output/tibia.assoc.txt",
                     as.is = "rs",header = TRUE)
head(gwscan)
nrow(gwscan)
```

Optional: Compare "genomic inflation" (part 2)

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

```
library(ggplot2)
library(cowplot) # Optional.
source("functions.R")
plot.inflation(gwscan$p_lrt)
```

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

- At this point, I recommend taking a screenshot of this plot before continuing.

Optional: Compare "genomic inflation" (part 3)

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

```
gwscan <-
  read.table("../output/tibia-lmm.assoc.txt",
             as.is = "rs",header = TRUE)
plot.inflation(gwscan$p_lrt)
```

Compare against the previous inflation plot—*did the LMM reduce inflation of p-values?*

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.
- However, LMMs can be computationally intensive, especially for data sets with many samples.
- Another issue: LMMs can *overcorrect* inflation—this is particularly a problem in mouse GWAS due to long-range LD patterns. One solution is a “leave-one-chromosome-out” strategy (this strategy was used in the Nicod *et al* paper).

Outline of workshop

1. Initial setup.
2. Download genotype data.
3. Prepare phenotype data.
4. Prepare genotype data.
5. Run a basic association analysis in GEMMA, and assess quality of results.
6. Run an LMM-based association analysis in GEMMA, and compare against the initial analysis.
7. **Visualize the genome-wide association signal in R to (a) map tibia loci and (b) identify plausible tibia genes.**

Visualizing the association signal

1. Summarize the genome-wide association signal for tibia length.
2. Take a closer look at the strongest association signal.
3. Based on the association signal, identify candidate genes.
4. *Optional:* Visualize the relationship between tibia length and the top SNP association.

Plot genome-wide scan for tibia length (part 1)

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:

```
getwd() # Should be .../code
gwscan <-
  read.table("../output/tibia-lmm.assoc.txt",
             as.is = "rs",header = TRUE)
head(gwscan)
nrow(gwscan)
```

Plot genome-wide scan for tibia length (part 2)

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

```
library(ggplot2)
library(cowplot) # Optional.
source("functions.R")
plot.gwscan(gwscan)
```

- *On which chromosomes do we observe the strongest association signals?*

Examine the association signal on chr. 6 (part 1)

Above, our plot gave us a *genome-wide* overview of the regions of the genome contributing to variation in BMD.

- Next, let's try to get a finer-scale view of the strongest association signal on chromosome 6.

Create a data frame with the chromsome 6 association results.

```
gwscan.chr6 <- subset(gwscan,chr == 6)
```

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

```
plot.region.pvalues(gwscan.chr6) +
  scale_x_continuous(breaks = seq(0,200,10))
```

Examine the association signal on chr. 6 (part 2)

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)
plot.region.pvalues(dat,size = 2) +
  scale_x_continuous(breaks = seq(145,150,0.5))
```

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

```
i <- which.min(gwscan.chr6$p_lrt)
gwscan.chr6[i,]
```

Examine the association signal on chr. 6 (part 3)

Based on the pattern of p-values, what would you propose as an approximate candidate region?

- *Exercise:* Use the UCSC Genome Browser (`http://genome.ucsc.edu`) to identify genes in or near the strongest association signal on chromosome 6 that may play a role in (tibia) bone development.
  - ▷ You will need to use NCBI release 38 (mm10) of the mouse genome assembly.

Optional: Visualize genotype-phenotype relationship

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

- *Here, let's look closely at the relationship between tibia length and the genotype at the top SNP.*

Load the GEMMA phenotype data:

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

Visualize genotype-phenotype relationship (part 2)

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

```
library(data.table)
geno <- fread("../data/chr06.geno.txt",sep = " ",
              stringsAsFactors = FALSE)
class(geno) <- "data.frame"
geno        <- round(geno[i,-(1:3)])
```

Note: We use `fread` from the `data.table` package because it is much, much faster than the base function `read.table`.

Visualize genotype-phenotype relationship (part 3)

If we convert the genotype to a "factor" (*i.e.*, categorical variable), we can use the boxplot function we used earlier:

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

I looked carefully at the SNP information and found that A is the alternative allele, so 2 corresponds to genotype AA.

```
print(draw.boxplot(geno,pheno) +
  labs(x = "genotype",y = "tibia length (mm)",
       title = "chr. 6: 146,795,232 bp"))
```

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

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

Observe that the "R-squared" (the effect size) is small.

Recap

- Today we implemented the basic steps of a genome-wide association analysis.
- We saw that most of the effort was getting the data ready for GEMMA, and carefully inspecting the GEMMA results.
- Did not discuss how to determine an appropriate significance threshold. This is a complicated question and beyond the scope of this workshop. **My advice:** first take an *exploratory approach* and identify the strongest association signals, then later assess significance.
- *Optional—but recommended—exercise:* Repeat the same data processing and analysis steps to map genetic loci for LDL concentrations in blood (units = mmol/L), using the `"Bioch.LDL"` column.