Admixture mapping

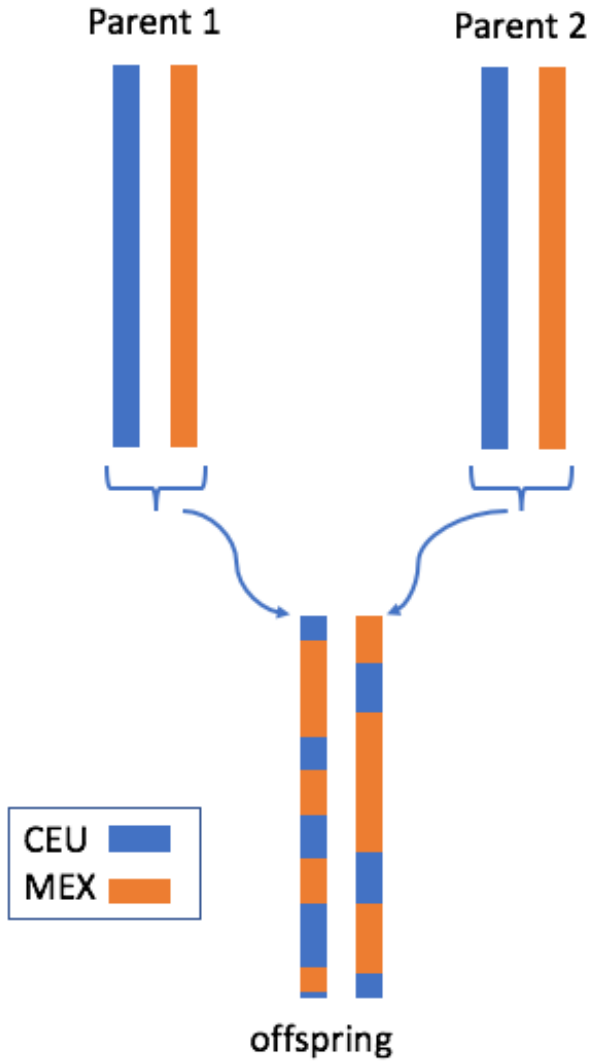
Tamar Sofer

July 2017

What is admixture mapping?

- Admixture mapping is a type of association analysis that can be performed only in admixed populations.
- In the history of mankind, sub-populations were formed, that have distinct genomic features.
  - Different frequencies of genetic variants, or population-specific genetic variants.
- Hispanics/Latinos are admixed.
  - A Hispanic/Latino individual likely have ancestors from three ancestral populations: European, African, Amerindian.
- Since chromosomes are inherited from parents in a process of recombination, for each person, segments of the chromosome were inherited from a specific ancestry (and ancestor).

What is admixture mapping?



What is admixture mapping?

- There are softwares that can delineate intervals along the chromosomes in a data set, and for each person, “say" which ancestry this interval was inherited from.
  - Software name: RFMix
  - It uses a “reference panel": genetic sequences from people in relevant ancestral populations.
  - Underlying local ancestry intervals (LAIs) are likely not exactly the same for all people, but they are defined the same way by the software.
- Local ancestry inference was performed in the HCHS/SOL by Browning et al. (2016, G3), and the results are available.

Looking at local ancestry files

- Let's look at a file with local ancestry information.
  - This is from the simulated data.

```
library("GWASTools")
dir <- paste0("/home/postdoc/tsofer/SISG/",
    "Preparing_simulated_data_2")
scanAnnot <- getobj(file.path(dir,
                    "SISG_phenotypes.RData"))
admixAnnot <- getobj(file.path(dir,
                    "SISG_local_ancestry_snpAnnot.RData"))
```

Looking at local ancestry files

```
dim(admixAnnot)
```

```
##       snps variables
##         75         6
```

```
head(pData(admixAnnot))
```

```
##   snpID pos.start pos.end position chromosome nmarkers
## 1     1    558390 1177226   867808          1       35
## 2     2   1177227 1805338  1491282          1       19
## 3     3   1805339 2108309  1956824          1       25
## 4     4   2108310 2272806  2190558          1       14
## 5     5   2272807 2412100  2342454          1       69
## 6     6   2412101 2737608  2574854          1       11
```

Looking at local ancestry files

- We need to use different functions (from before) to read the data
- Because now we have genotype dosages/count per ancestry!

```
require(gdsfmt)
gds <- openfn.gds(file.path(dir,
                            "SISG_local_ancestry_2.gds"))
gds
```

```
## File: /home/postdoc/tsofer/SISG/Preparing_simulated_data
## +    [  ]
## |--+ CEU_dosage   { Bit2 75x500, 9.2K }
## |--+ MEX_dosage   { Bit2 75x500, 9.2K }
## |--+ sample.id   { Str8 500, 2.3K }
## |--+ snp.id   { Int32 75, 300B }
## |--+ snp.chromosome   { Float64 75, 600B }
## \--+ snp.position   { Int32 75, 300B }
```

Looking at local ancestry files

```
ancestries <- c("CEU", "MEX")
genoDataList <- vector(mode = "list", length = 2)
names(genoDataList) <- ancestries
for (ancestry in ancestries){
  gds.reader <- GdsGenotypeReader(gds,
                genotypeVar=paste0(ancestry, "_dosage"))

  genoDataList[[ancestry]] <- GenotypeData(gds.reader,
                                           scanAnnot=scanAnnot)
}
```

Looking at local ancestry files

- Now we have a list with genotype readers.
- We can apply functions presented before on each of the list components.

```
genoDataList[["CEU"]]
```

```
## An object of class GenotypeData
##   | data:
## File: /home/postdoc/tsofer/SISG/Preparing_simulated_data
## +    [  ]
## |--+ CEU_dosage   { Bit2 75x500, 9.2K }
## |--+ MEX_dosage   { Bit2 75x500, 9.2K }
## |--+ sample.id   { Str8 500, 2.3K }
## |--+ snp.id   { Int32 75, 300B }
## |--+ snp.chromosome   { Float64 75, 600B }
## \--+ snp.position   { Int32 75, 300B }
##   | SNP Annotation:
## NULL
```

Looking at local ancestry files

- Now we have a list with genotype readers. We can apply the same function as before on each of the list components.

```
getGenotype(genoDataList$CEU)[1:4,1:4]
```

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

```
getScanID(genoDataList$MEX)[1:4]
```

```
## [1] "p1" "p2" "p3" "p4"
```

```
## Use one list slot to close the connection to file-

close(genoDataList[[1]])
```

Admixture mapping - how does it work?

- The local ancestry "genotypes" are not actually counts of genotypes.
- They are counts of ancestry!
- So if person `p1` have count of 2 for its CEU dosage in interval 3, it means that
  - Interval 3 in the `p1`'s two chromosomes 1 was inherited from a CEU ancestor.
- In admixture mapping, we test the local ancestry counts.

Admixture mapping - how does it work?

- In admixture mapping, we test the local ancestry counts.
- Why is it meaningful? intuition:
  - Because if a genotype is more frequent in one population (pop1);
  - And it is also associated with a trait;
  - People with more of the intervals spanning the genotypes inherited from pop1, are more likely to have the trait.
- Mathematically, it can be shown that
  - If there are two ancestral populations;
  - There is a single causal variant in an interval;
  - The genotype effect size β is the same in the two populations;
  - The allele frequency in the two ancestral populations are f_1 and f_2;
  - Then the effect size estimated by testing the LAI counts is $\beta(f_1 - f_2)$.

Admixture mapping - how does it work?

- The effect size estimated by testing the LAI counts is $\beta(f_1 - f_2)$.
  - This is lower than the effect size of the genotype β.
  - If $f_1 == f_2$, this "effect" equals zero, and so admixture mapping is not useful.
  - If $f_1 = 0$ or $f_2 = 0$, it'll equal $f_2\beta$ or $f_1\beta$.
  - Regular association mapping may actually give something smaller.

Admixture mapping power

- Suppose we had a specific variant g genotyped on all peopled.
- And we had the local ancestry counts in the interval spanning g.
- Then, testing the genotype association will be more powerful than testing the local ancestry.

However...

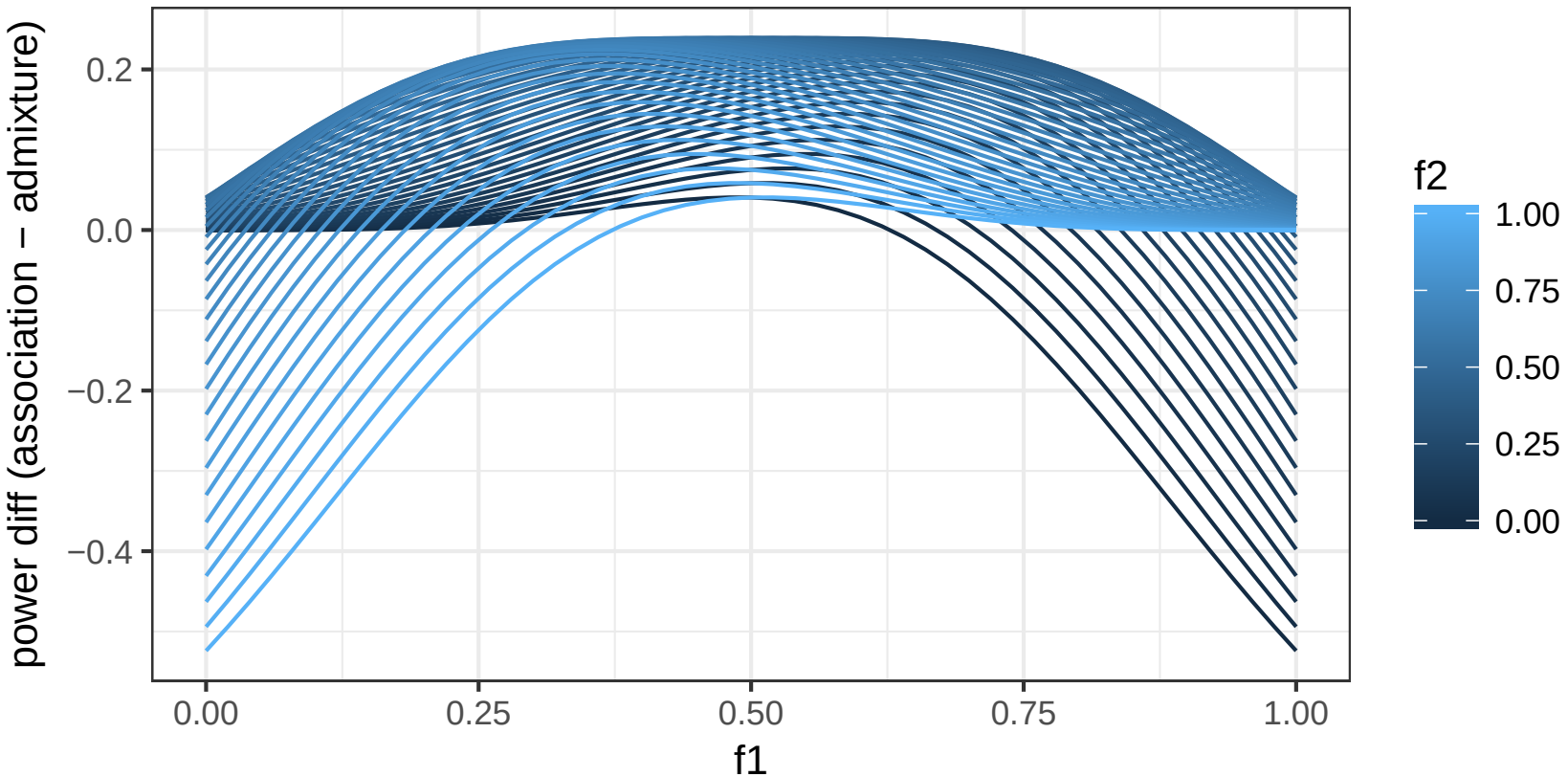
- Sometimes we don't have g genotyped, but we do have ancestry counts for an interval spanning g.
- There are many less local ancestry intervals than genotypes
  - Reducing the p-value required for significance of LAI associations.
- The causal g may be rare, making genotype associations unstable, even if the genotype is in the data.
  - While LAI associations do not suffer from this instability.

Admixture mapping is useful in these cases.

Admixture mapping power

- Consider the case where we do have the causal genotype.
- Its effect $\beta = 0.06$, and the frequency of the local ancestry from pop1 is 0.5.
- Significance of association results: 5×10^{-8}, of admixture mapping results: 5.7×10^{-5}.
- See figure for difference in power comparing association to admixture mapping as a function of ancestry-specific frequencies.

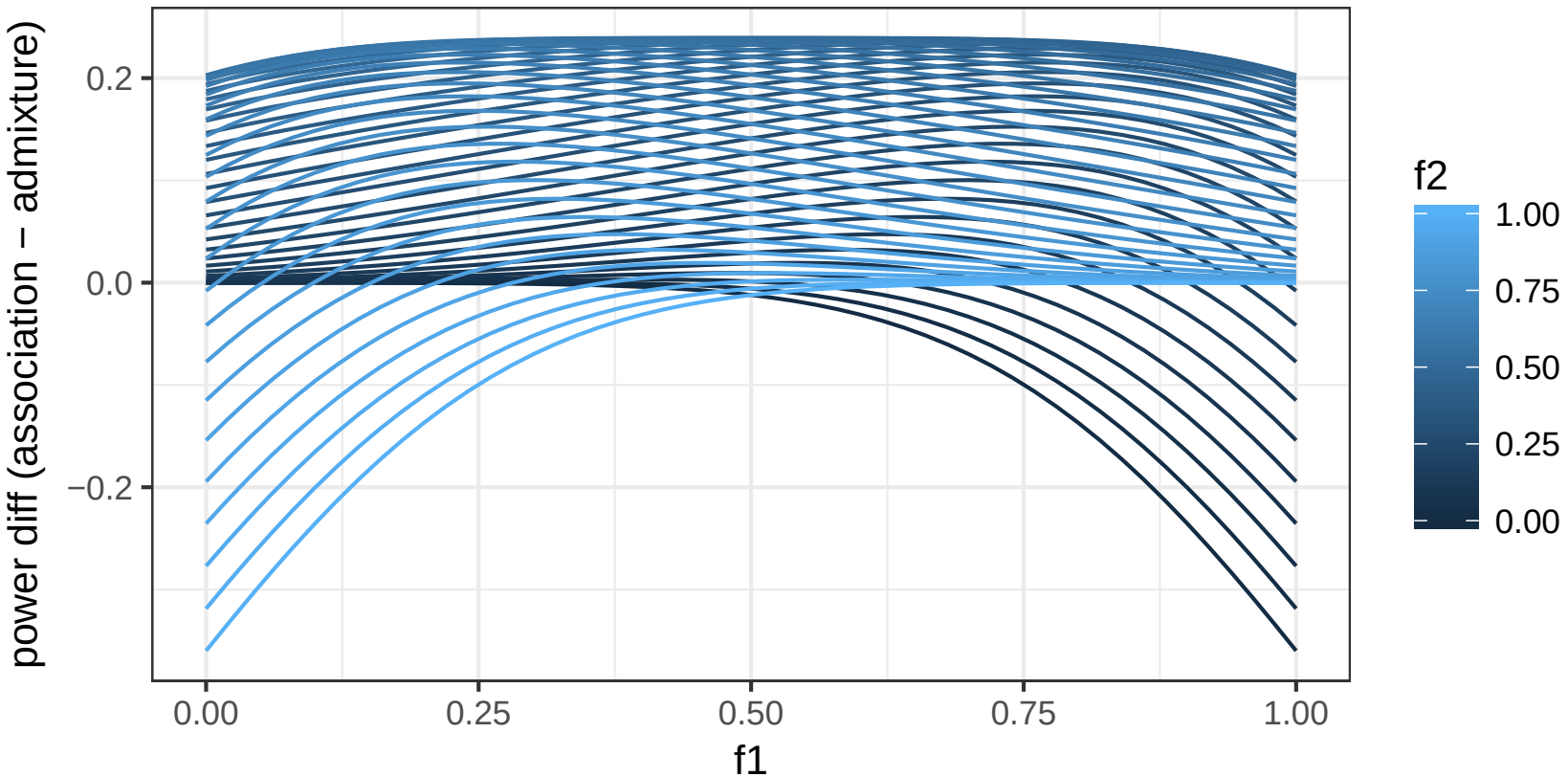
Admixture mapping power



Admixture mapping power

- Causal genotype effect $\beta = 0.06$, and the frequency of the local ancestry from pop1 is lower, 0.2.
- Significance of association results: 5×10^{-8}, of admixture mapping results: 5.7×10^{-6}
- Comparing the previous figure to the next, shows less advantage for admixture mapping in this case.

Admixture mapping power



Admixture mapping power

Additional settings in which admixture mapping is useful:

- The effect sizes is different between ancestries (even if the allele frequencies are the same across ancestries!).
- Multiple causal variants in the LAI.

Take-home message: admixture mapping can sometimes detect association regions that association mapping cannot.

Admixture mapping - let'd do it!

- First, let's open again the connection to file
- And create genotypeReaders.

```
gds <- openfn.gds(file.path(dir,
                    "SISG_local_ancestry_2.gds"))
ancestries <- c("CEU", "MEX")
genoDataList <- vector(mode = "list", length = 2)
names(genoDataList) <- ancestries
for (ancestry in ancestries){
  gds.reader <- GdsGenotypeReader(gds,
              genotypeVar=paste0(ancestry, "_dosage"))

  genoDataList[[ancestry]] <- GenotypeData(gds.reader,
                                    scanAnnot=scanAnnot)
}
```

Admixture mapping - let'd do it!

- Prepare for analysis...

```
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)
```

Admixture mapping - let'd do it!

- Fit the null mixed model as before,
- and test using the GENESIS function admixMapMM.(MM: mixed model)

```
library(GENESIS)
nullmod <- fitNullMM(scanData = scanAnnot,
                     outcome = outcome,
                     covars = covariates,
                     covMatList = covMatList,
                     verbose = FALSE)
# run admixture mapping
assoc.admix <- admixMapMM(genoDataList,
                          nullMMobj = nullmod)
```

```
## Running analysis with 500 Samples and 75 SNPs

## Beginning Calculations...

## Block 1 of 1 Completed = 0.04685 secs
```

Admixture mapping - let'd do it!

```
close(genoDataList[[1]])
head(assoc.admix)
```

```
##   snpID chr   n CEU.freq CEU.Est   CEU.SE MEX.freq MEX.P
## 1     1   1 500    0.691      NA       NA    0.309
## 2     2   1 500    0.686   3.500 54862950    0.314   4.5
## 3     3   1 500    0.677      NA       NA    0.323
## 4     4   1 500    0.707      NA       NA    0.293
## 5     5   1 500    0.703   1.625 47546687    0.297   1.3
## 6     6   1 500    0.721      NA       NA    0.279
##   Joint.Stat Joint.pval
## 1         NA         NA
## 2 0.93551254  0.6264062
## 3         NA         NA
## 4         NA         NA
## 5 0.05489418  0.9729262
## 6         NA         NA
```

Admixture mapping - let'd do it!

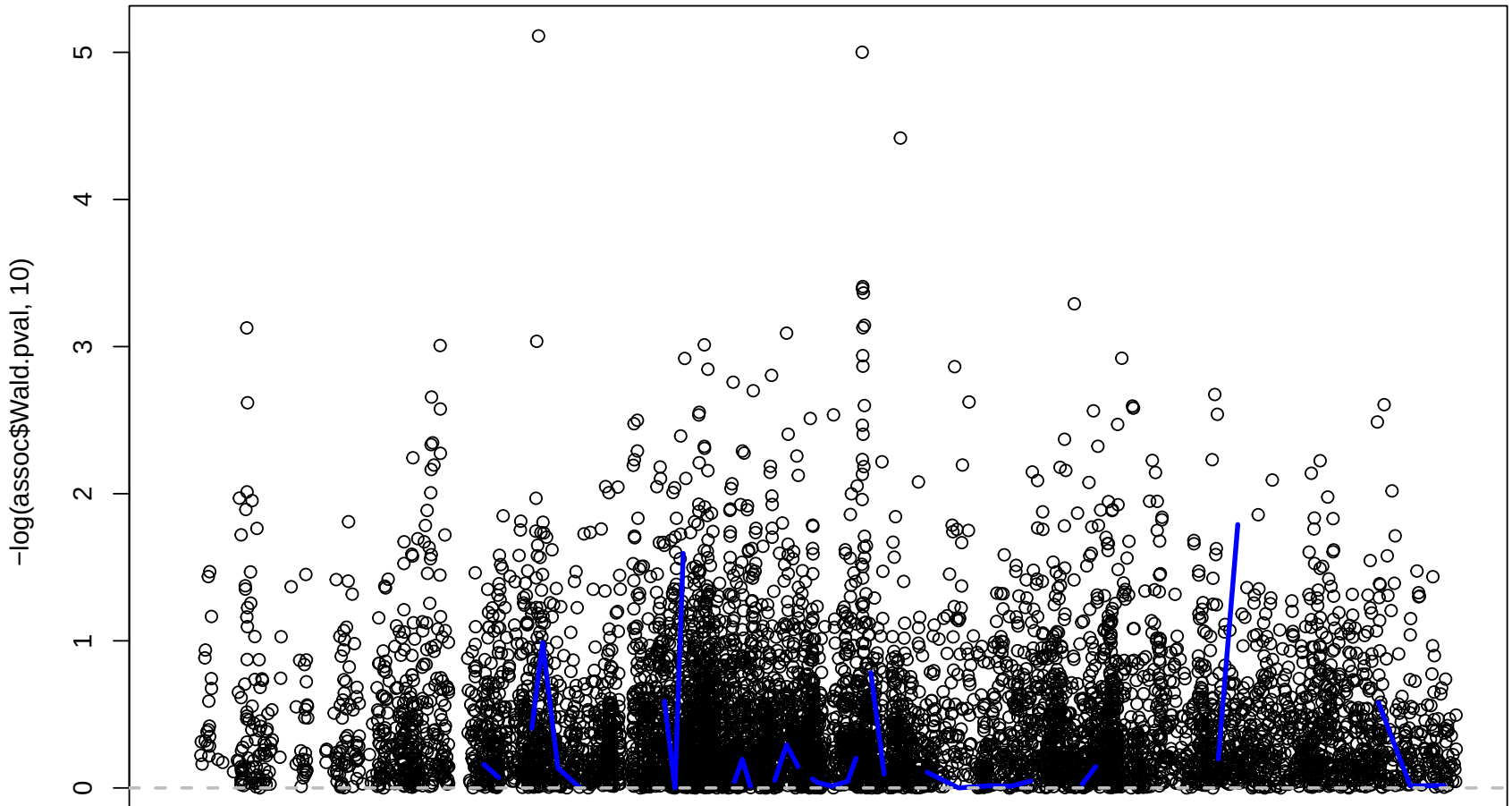
- Let's also perform the usual association testing.

```
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  , nullMMobj = nullmod)
```

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

## Beginning Calculations...

## Block 1 of 2 Completed - 1.562 secs

## Block 2 of 2 Completed - 0.5801 secs
```

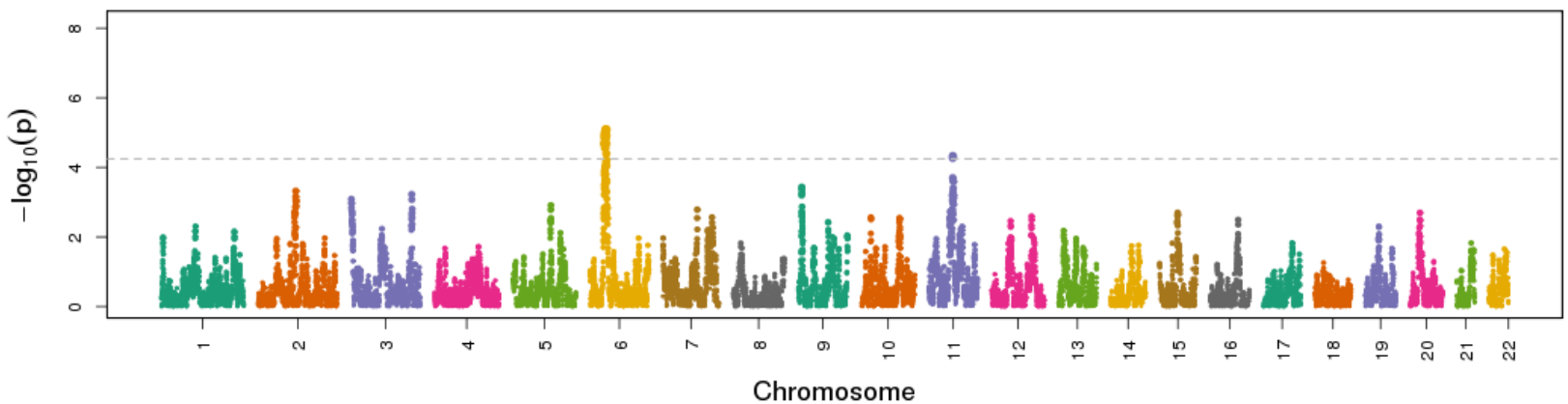
Admixture mapping - let'd do it!

```
plot(snpAnnot$position, -log(assoc$Wald.pval, 10))
lines(admixAnnot$position,
        -log(assoc.admix$Joint.pval, 10),
        col = "blue", lwd = 3)
abline(h = 5.7e-5, col = "grey", lty = 2, lwd = 2)
```



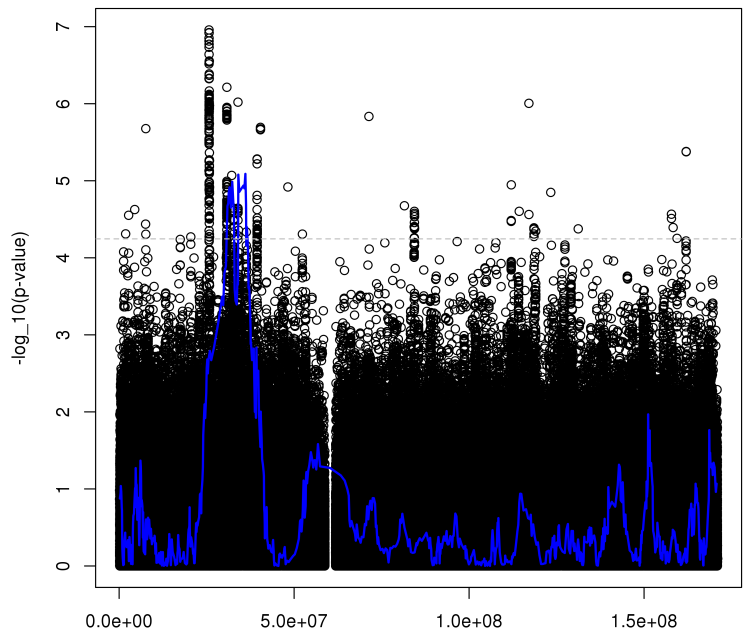
Example from our work: admixture mapping of mean arterial pressure (MAP)

- We performed admixture mapping of blood pressure traits (under review).
- Analysis of MAP. Here is the Manhattan plot from the MAP analysis, testing counts of LAI from Amerindian ancestry (baseline is either European or African).



Example from our work: admixture mapping of mean arterial pressure (MAP)

- Here's a figure with the significant admixture mapping region on chromosome 6.
- With the usual association analysis results (from GWAS that tested SNP allele dosages).
- Question: why do we use symbols when testing SNPs, lines when testing LAIs?



Example from our work: admixture mapping of mean arterial pressure (MAP)

- Our end goal is to find genetic variants driving the association.
- How can we find them?

Recall: admixture mapping associations are detected when there is a difference in allele frequencies between ancestries. Also, when there is a difference in ancestry-specific effect sizes.

Example from our work: admixture mapping of mean arterial pressure (MAP)

Candidate variants explaining the admixture mapping are likely:

- Somewhat significant in usual association testing.
- Have different frequencies between ancestry.
  - Or different effect sizes.

But we don't have ancestry-specific frequencies, nor effect sizes!

- Proxy: different genetic analysis groups have different ancestry proportions.
- difference in ancestry-specific frequencies, likely lead to differences in allele frequencies between genetic analysis groups!

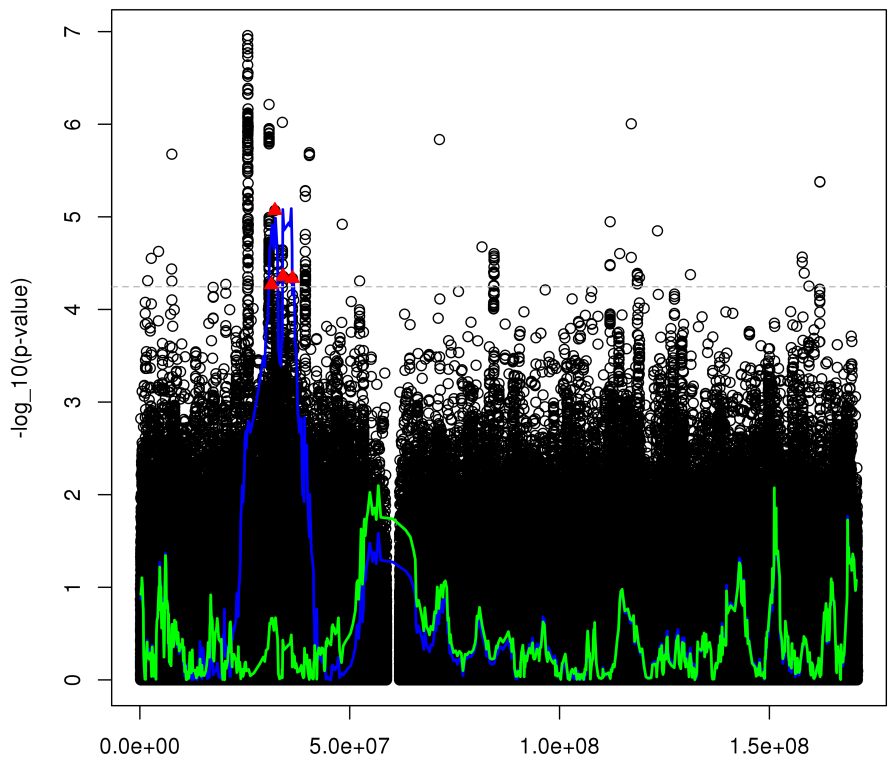
Example from our work: admixture mapping of mean arterial pressure (MAP)

- We identified a set of SNP candidate that were
  - Somewhat significant in association testing
  - Had difference in allele frequencies between the Mexican (high Amerindian ancestry proportion) and Cuban (low) genetic analysis group.
- And LD pruned them, to get four SNPs.

A1 freq Mexican	A1 freq Cuban	Effect	SE	p-value
0.80	0.68	-0.72	0.18	5.40E-05
0.99	0.99	-3.19	0.72	8.53E-06
0.68	0.94	0.82	0.20	4.39E-05
0.81	0.98	1.00	0.25	4.62E-05

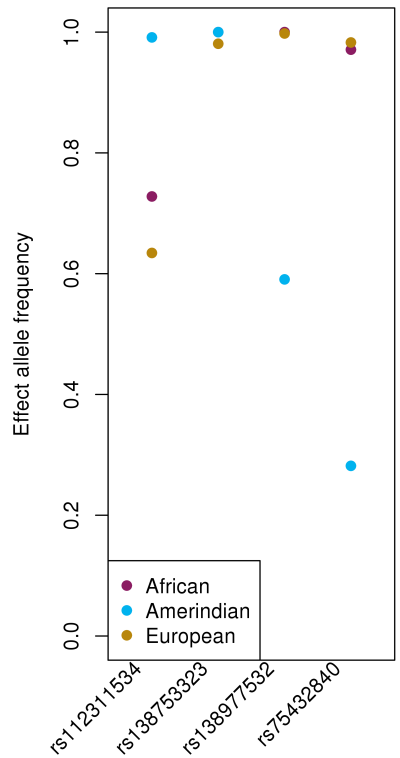
Example from our work: admixture mapping of mean arterial pressure (MAP)

- Next: check whether these SNPs drive the admixture mapping association.
- Perform conditional analysis - the same linear mixed model, with these SNPs as covariates.



Example from our work: admixture mapping of mean arterial pressure (MAP)

- Finally, we can also use the package ASAFE to calculate ancestry-specific allele frequencies of the four SNPs.



Exercises

1. Which is the LAI with the most significant p-value in the admixture mapping?
   - How many genotyped/imputed variants are in this LAI?
   - What is the most significant SNP in this LAI?
   - Can you use the formula $\beta(f_1 - f_2)$ to guestimate an effect size for the potential causal SNP in the LAI? if yes, what is it? if not, why not?
2. Find the genotype with the largest MAF difference between the UW and UNC groups
3. Use this genotype in simulating a new trait with the code in the next slide, and run a new admixture mapping analysis.

Code for simulations

- First we find a SNP with large MAF differences between the UNC and UW groups.

```
gds.geno <- GdsGenotypeReader(file.path(dir,
                              "SISG_snp_dosages.gds"))
genoData <- GenotypeData(gds.geno,
                  snpAnnot=snpAnnot, scanAnnot = scanAnnot)
Afreqs.unc <- alleleFrequency(genoData, scan.exclude =
                      scanAnnot$scanID[
                          which(scanAnnot$group == "uw")],
                      verbose = FALSE)
Afreqs.uw <- alleleFrequency(genoData, scan.exclude =
                      scanAnnot$scanID[
                          which(scanAnnot$group == "unc")],
                      verbose = FALSE)
```

Code for simulations

- Extract the genotypes

```
select.genotype.ind <- which.max(
      abs(Afreqs.unc[,"MAF"] - Afreqs.uw[,"MAF"]))
Afreqs.uw[select.genotype.ind, "MAF"]
```

```
## [1] 0.33
```

```
Afreqs.unc[select.genotype.ind, "MAF"]
```

```
## [1] 0.2233333
```

```
genotype.id <- getSnpID(gds.geno)[select.genotype.ind]
causal.genotype <- getGenotypeSelection(gds.geno,
                                        snpID = genotype.id)
close(gds.geno)
```

Code for simulations

- Use the genotype in simulations

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

- Now run admixture mapping with the outcome "new.trait"!
- Run also association analysis, and create a figure with the results on top of each other.