GWAS Quality Control

Randy Johnson

3/30/2017

Setup

```
knitr::opts_chunk$set(echo = FALSE)
library(GWASTools)
library(SNPRelate)
library(tidyverse)
library(cowplot)
library(hwde)
library(ggplot2)

# colorbline palette
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

Quality Control

- Aim of quality control: reduce bias in our statistics.
- Many of the methods discussed today apply to other types of analysis.

Microarray vs NGS Technologies

	The speed of microarrays	+ The power of next-gen sequencing
Whole-Genome Analysis	Whole-Genome SNP Genotyping	Whole-Genome SNP Discovery
Copy Number Variation (CNV)	CNV Analysis	CNV Discovery
Targeted Genome Analysis	Custom and Focused SNP Genotyping	Targeted Resequencing
Gene Regulation and Epigenetic Analysis	Whole-Genome DNA Methylation Profiling	• Whole-Genome DNA Methylation Discovery and Analysis • Chromatin Immunoprecipitation (ChIP-Seq) • Small RNA Discovery and Analysis
Gene Expression	• Whole-Genome Gene Expression Analysis • FFPE Sample Analysis	Transcriptome Discovery and Profiling
Cytogenetics	Cytogenetic Abnormalities	Digital Karyotyping

Figure 1: Advantages of using microarray vs Next-Generation Sequencing (NGS) technologies (Caroline Thureau, 2010)

Summary of How Microarrays Work: Affymetrix

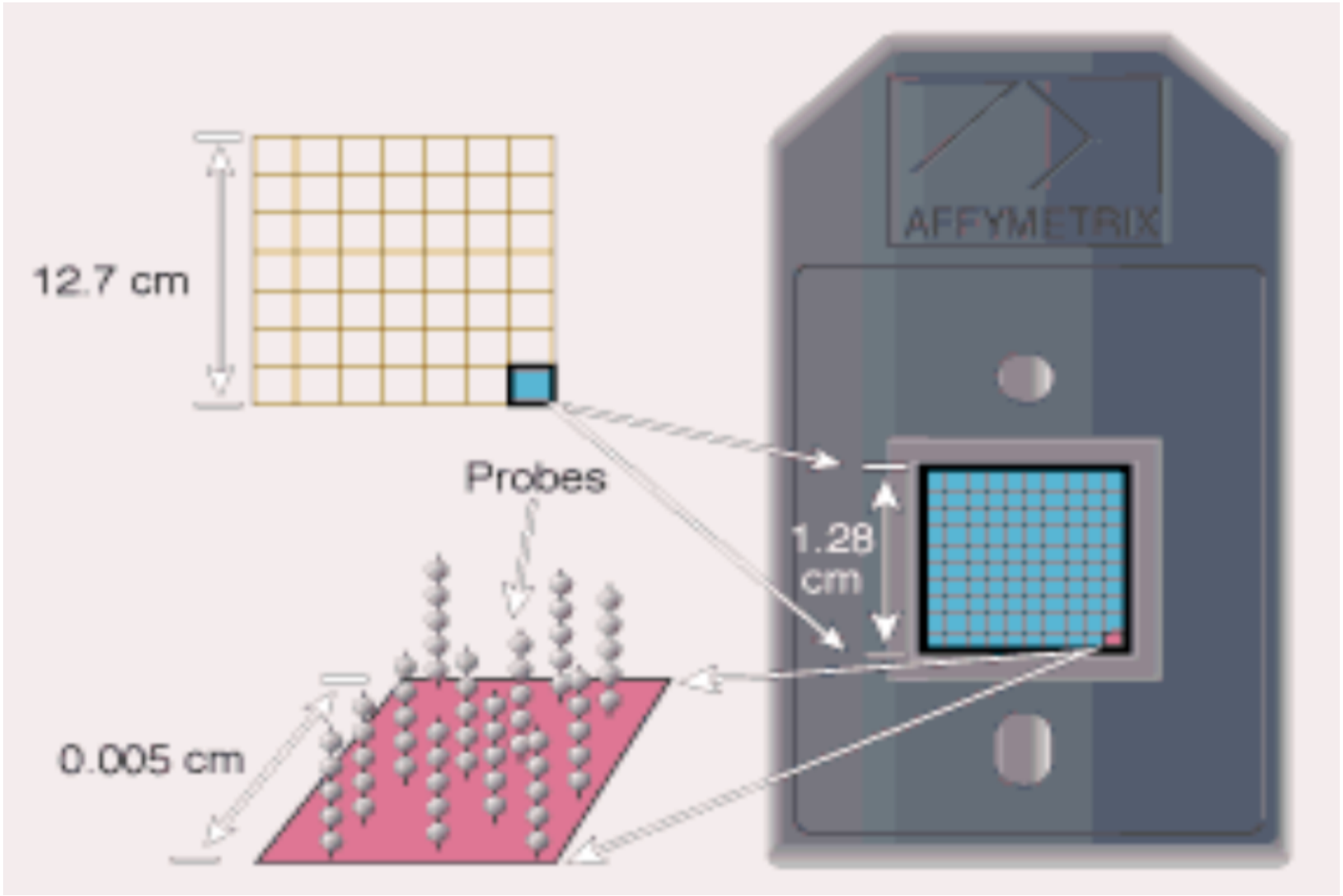


Figure 2: The Affymetrix platform (image courtesy of Nickerson)

Summary of How Microarrays Work: Affymetrix

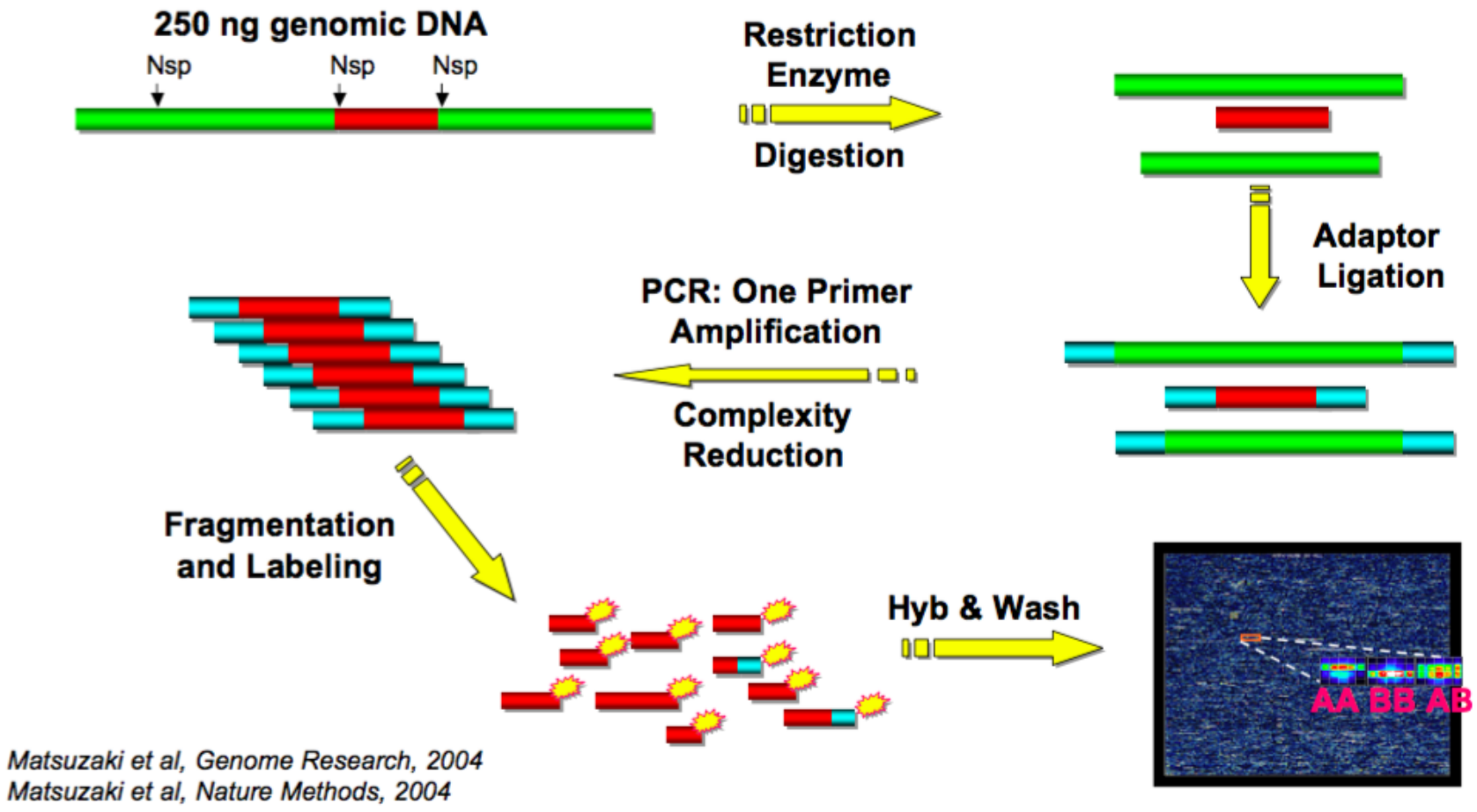


Figure 3: Affymetrix Workflow: Preparation of samples for Illumina chips is similar (image courtesy of Nickerson)

Summary of How Microarrays Work: Illumina

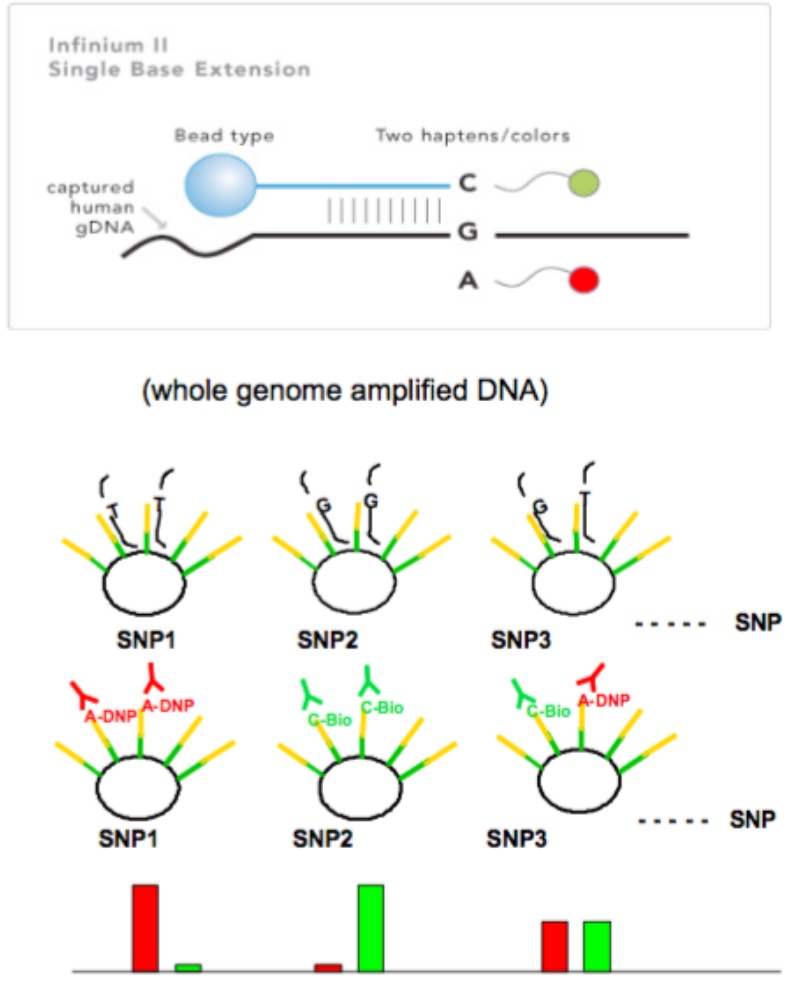


Figure 4: DNA fragments are captured on microbeads in Illumin platform (image courtesy of Nickerson)

Summary of How Microarrays Work: Illumina

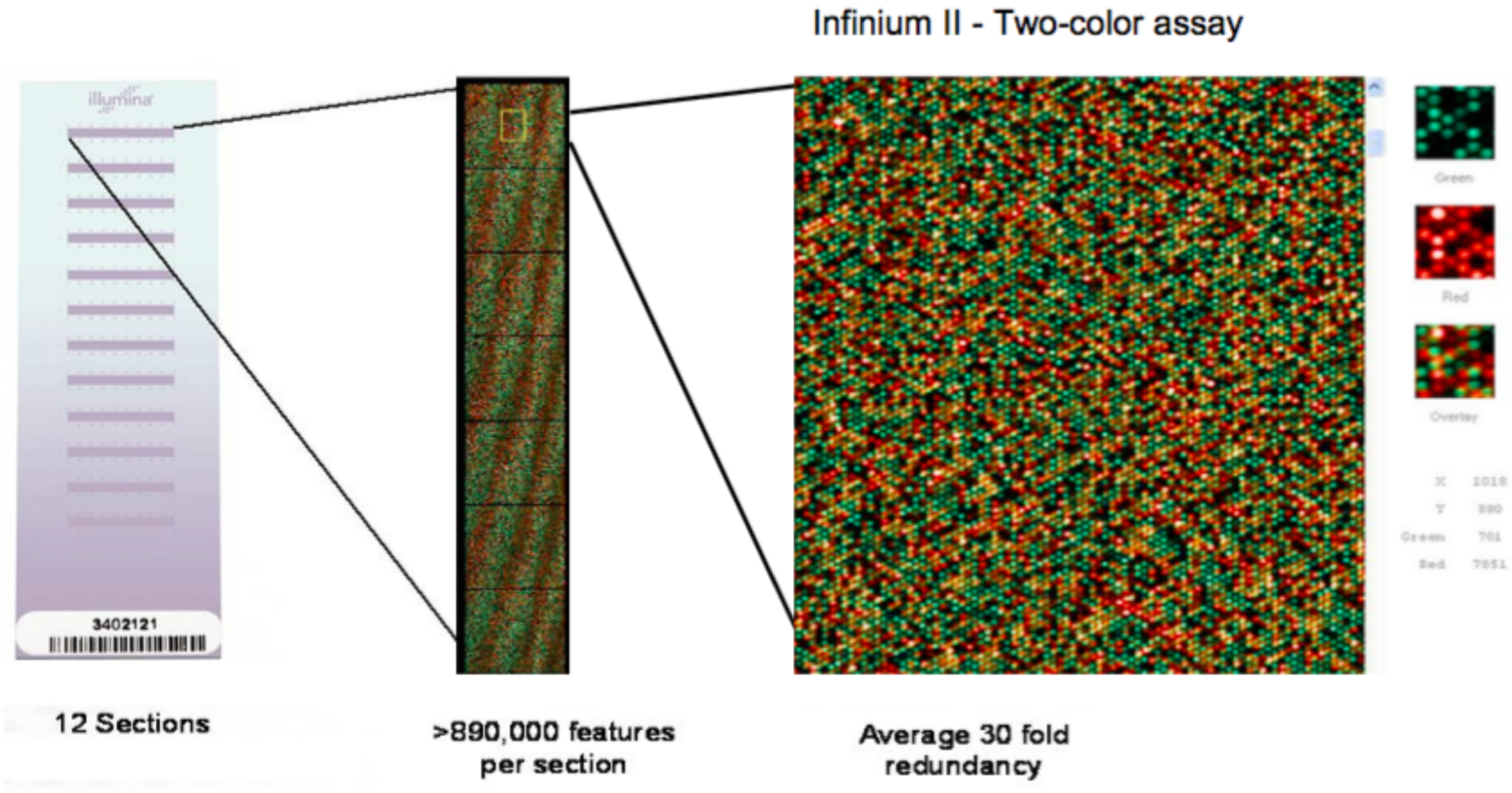


Figure 5: Beads are randomly dispersed on a plate and later decoded. Redundancy is higher (image courtesy of Nickerson)

Summary of How Microarrays Work: Genotype Inferrence

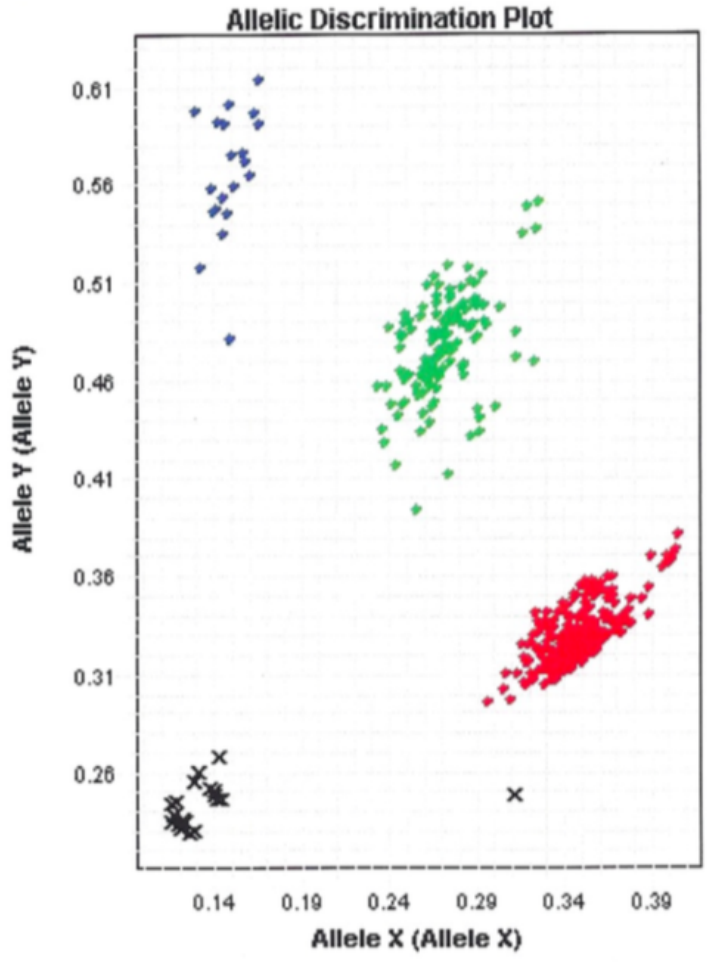


Figure 6: Intensities of A and B alleles are clustered to infer genoyptes.

Genome Wide Association Studies (GWAS)

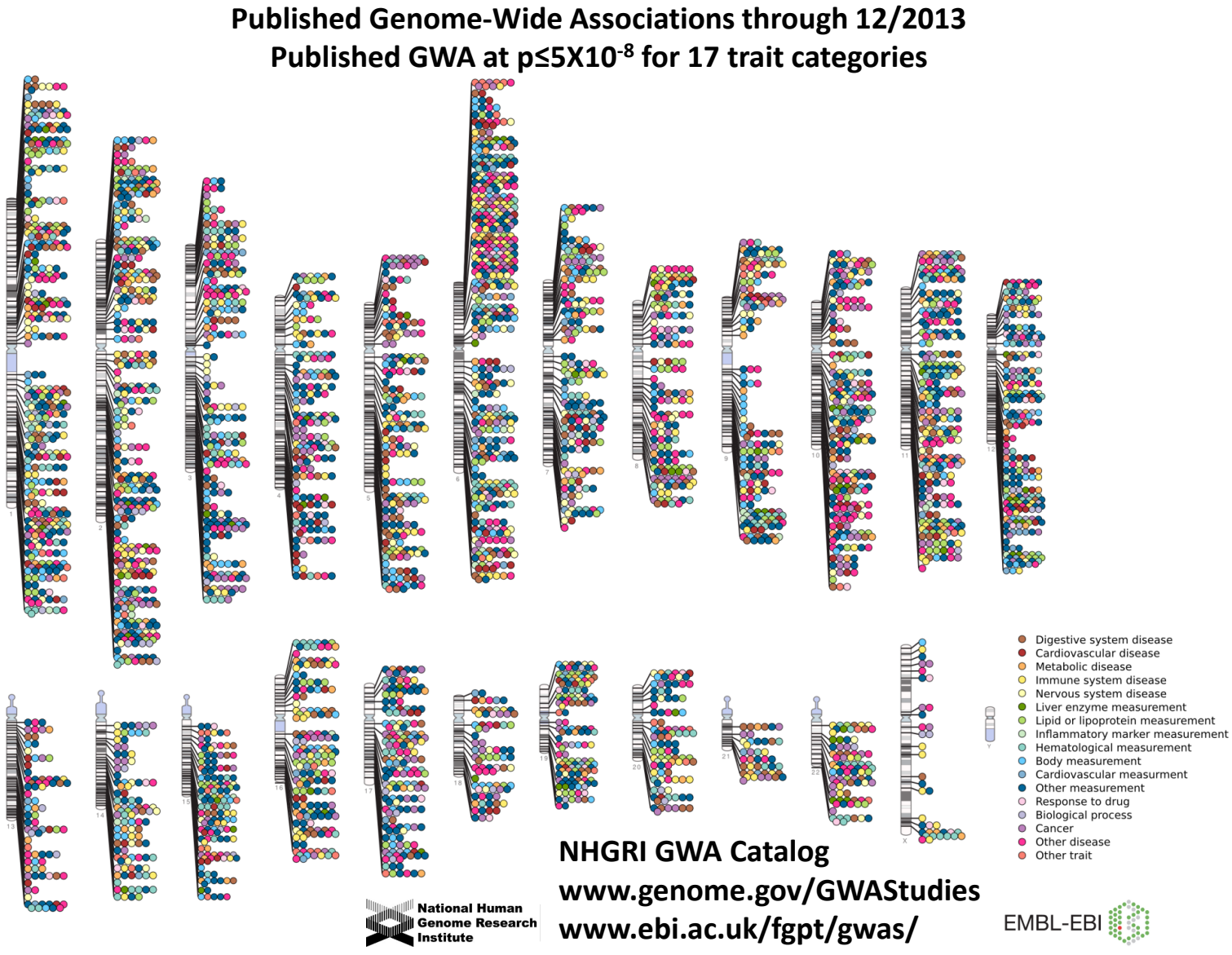


Figure 7: GWAS Findings as of 2013 (Hindorf et. al.)

Batch Effects: Problem

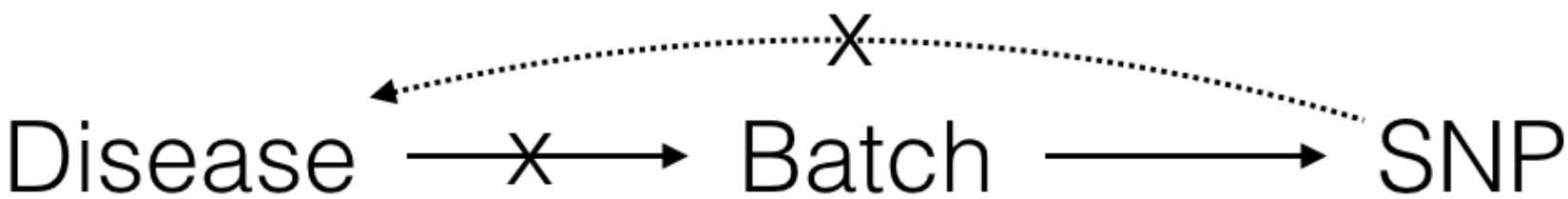
- Microarrays are tricky things
- Results can be slightly influenced (i.e. different) by processing date, lab tech, equipment used, ...
- Every effort should be made to follow your protocol as exactly as possible, but you are still going to end up with batch effects.

Figure 8: Batch Effect Bias

Batch Effects: Solution

- Solution: Randomization breaks the link between disease and batch, which breaks the false association between SNP and disease.
- Alternate solution: Include batch number in your statistical model. Randomization is better, but sometimes we don't get a say in the matter.

$$\begin{aligned} \textit{disease} =& \beta_0 + \beta_{b1}\textit{batch}_1 + \beta_{b2}\textit{batch}_2 + \cdots + \\ & \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon \end{aligned}$$



Batch Randomization

Lets say we have 1000 samples we need to split over 10 batches of 100 each.

```
set.seed(87346)
samples <- data_frame(
                id = 1:1000,
                disease = c(rep(TRUE, 500), rep(FALSE, 500)),
                ## other clinical observations,
                batch = sample(rep(1:10, each = 100), 1000))
```

Batch Randomization

Again, our study will collect 1000 samples, which will need to be split into 10 batches of 100 each. This time, however, we expect the samples to be collected over a 5 year period, due to the low incidence of our disease. Also, we can't wait until the end of collection to run all the batches at once.

- Collect controls at the same time as the cases!!
- When you have collected 50 cases with their corresponding 50 controls, run a batch.
- If your technology has sub-batches (e.g. lanes) randomize the distribution of your cases and controls across those sub-batches as well.

Genotyping Quality: Problem

- We assume that the technology is performing as it should.
- If it is performing as it should, we expect to get complete data (i.e. not very many missing values).
- If there are too many missing values, something is probably wrong with either the sample or the technology!
- Sources of "wrongness" include:
  - Poor DNA quality
  - Poor reagent quality
  - Contamination
  - Poor adherance to protocol

Genotyping Quality: Solution

If there is a problem with your genotyping quality, you are really only left with two viable options:

- Redo the genotyping
- Remove the offending individuals/SNPs from the analysis

You can expect a small proportion of your SNPs to fail, and it is not uncommon to have a small proportion of your samples fail. The general rule of thumb is: there should be no more than 3-5% of your genotypes are missing for each individual, and no more than 3-5% of individuals have a missing genotype for each SNP.

- Too many missing genotypes for a SNP: Assay problem
- Too many missing genotypes for an individual: Sample problem

Genotyping Quality: Non-Solution

- Fancy bioinformatics will not solve genotyping quality problems!

- If your DNA is of low quality, your data are of low quality.
- If your assay is of low quality (e.g. due to a bad lot of a reagent), your data are of low quality.

- If your data are of low quality, your statistics are suspect.

- If your statistics are suspect, your inferences/conclusions are also suspect (i.e. they are no good)!

Genotyping Quality: Solution

- This will remove all SNPs with more than 5% of their genotypes missing (you may also want to remove any SNPs with a very low minor allele frequency using the `--maf` option):

```
plink --bfile mydata --geno 0.05 --maf 0.01 --recode
```

- This will remove all individuals with more than 5% of their genotypes missing:

```
plink --bfile mydata --mind 0.05 --recode
```

- The `--recode` option will generate a new set of PLINK files. You can include more than one filtering command in a single call.

Genotyping Quality: HWE

- Hardy-Weinberg Equilibrium (HWE) is another indicator of poor genotyping quality.

	A	a
A	f_{AA}	f_{aA}
a	f_{Aa}	f_{aa}

$$p = f_{AA} + \frac{f_{Aa} + f_{aA}}{2}$$

$$q = f_{aa} + \frac{f_{Aa} + f_{aA}}{2}$$

HWE: Assumption of Genetic Equilibrium

Under the assumption of genetic equilibrium in the population,

$$
\begin{aligned}
f_{AA} &= \mathrm{P}(A \text{ from mom} \cap A \text{ from dad}) \\
&= p^2, \\
f_{aa} &= \mathrm{P}(a \text{ from mom} \cap a \text{ from dad}) \\
&= q^2, \\
f_{Aa} &= f_{aA}, \\
f_{Aa} + f_{aA} &= \mathrm{P}(A \text{ from mom} \cap a \text{ from dad}) + \\
&\quad \mathrm{P}(a \text{ from mom} \cap A \text{ from dad}) \\
&= 2pq, \\
p^2 + 2pq + q^2 &= 1.
\end{aligned}
$$

Genotyping Quality: Testing HWE Assumption

We can test this assumption using a Chi-squared test as follows:

- Calculate p and q (sample size is n).
- Define:
  - O_{AA} = Observed number of individuals with AA genotype,
  - O_{Aa} = Observed number of individuals with Aa genotype,
  - O_{aa} = Observed number of individuals with aa genotype.

$$\frac{(O_{AA} - n * p^2)^2}{n * p^2} + \frac{(O_{Aa} - n * 2pq)^2}{n * 2pq} + \frac{(O_{aa} - n * q^2)^2}{n * q^2} \sim \chi_1^2$$

- Some SNPs will not fall within HWE expectations simply as a natural result of frequency deviations
- Research indicates that most of these deviations are due to genotyping errors

Genotyping Quality: HWE Assumption Example 1

$$
\begin{aligned}
O_{AA} &= 1469 \\
O_{Aa} &= 138 \\
O_{aa} &= 5 \\
n &= 1469 + 138 + 5 \\
&= 1612 \\
p &= \frac{2 * O_{AA} + O_{Aa}}{2n} \\
&= \frac{2 * 1469 + 138}{2 * 1612} \\
&= 0.954 \\
q &= \frac{2 * O_{aa} + O_{Aa}}{2n} \\
&= \frac{2 * 5 + 138}{2 * 1612} \\
&= 0.046
\end{aligned}
$$

$$
\begin{aligned}
q &= (1 - p) \\
E_{AA} &= n * p^2 \\
&= 1612 * 0.954^2 \\
&= 1467.4 \\
E_{Aa} &= n * 2pq \\
&= 1612 * (2 * 0.954 * 0.046) \\
&= 141.2 \\
E_{aa} &= n * q^2 \\
&= 1612 * 0.046^2 \\
&= 3.4 \\
\chi_1^2 &= \sum \frac{(O - E)^2}{E}
\end{aligned}
$$

Genotyping Quality: HWE Assumption Example 1

$$\begin{aligned}\chi^2_1 &= \frac{(1469 - 1467.4)^2}{1467.4} + \frac{(138 - 141.2)^2}{141.2} + \frac{(5 - 3.4)^2}{3.4} \\ &= 0.001 + 0.073 + 0.756 \\ &= 0.83\end{aligned}$$

```
pchisq(0.83, 1, lower.tail = FALSE)
```

```
## [1] 0.3622725
```

```
hwexact(1469, 138, 5)
```

```
## [1] 0.3825187
```

Genotyping Quality: HWE Assumption Example 2

$$
\begin{aligned}
O_{AA} &= 1465 \\
O_{Aa} &= 138 \\
O_{aa} &= 9 \\
n &= 1465 + 138 + 9 \\
&= 1612 \\
p &= \frac{2 * O_{AA} + O_{Aa}}{2n} \\
&= \\
&= \\
q &= \frac{2 * O_{aa} + O_{Aa}}{2n} \\
&= \\
&= \\
q &= (1 - p) \\
E_{AA} &= n * p^2 \\
&= \\
&= \\
E_{Aa} &= n * 2pq \\
&= \\
&= \\
E_{aa} &= n * q^2 \\
&= \\
&= \\
\chi_1^2 &= \sum \frac{(O - E)^2}{E}
\end{aligned}
$$

Genotyping Quality: HWE Assumption Example 2

$$\chi_1^2 = \frac{\qquad}{\qquad} + \frac{\qquad}{\qquad} + \frac{\qquad}{\qquad}$$

$$= \qquad$$

$$= \qquad$$

```
pchisq(      , 1, lower.tail = FALSE)
hwexact(     ,      ,    )
```

$$p_{\chi^2} =$$

$$p_{exact} =$$

Genotyping Quality: HWE Solution

Use the `--hwe` option in PLINK to filter on a specific threshold (0.001 is common),

```
plink --bfile mydata --hwe 0.001 --recode
```

or you can get a summary of the HWE test statistics using the `--hardy` option,

```
plink --bfile mydata --hardy
```

which creates a file called `plink.hwe` with the following columns of data:

```
## SNP              SNP identifier
## TEST             Code indicating sample
## A1               Minor allele code
## A2               Major allele code
## GENO             Genotype counts: 11/12/22
## O(HET)           Observed heterozygosity
## E(HET)           Expected heterozygosity
## P                H-W p-value
```

Aside on the χ^2 Test (i.e. Pearson χ^2 Test)

- This is used to test the frequency distribution of a sample is consistent with a theoretical distribution.
  - NULL under HWE: $E_{AA} = np^2$, $E_{AB} = 2npq$, and $E_{BB} = nq^2$.
  - NULL for 2 tosses of fair coin (n times): $E_{HH} = 0.25n$, $E_{HT} = 0.5n$, $E_{TT} = 0.25n$.
  - NULL for 1 toss of fiar die (n times): $E_1 = E_2 = \cdots = E_6 \approx 0.167n$.
- The test statistic takes the form:

$$\chi_1^2 = \sum_j \frac{(O_j - E_j)^2}{E_j}$$

Asside on the χ^2 Test: Example

- Our test statistic for the number of coin tosses is:

$$\chi_1^2 = \frac{(O_{HH} - 0.25n)^2}{0.25n} + \frac{(O_{HT} - 0.5n)^2}{0.5n} + \frac{(O_{TT} - 0.25n)^2}{0.25n}$$

Aside on the χ^2 Test: Example

```
set.seed(872364)
n <- 100
tosses <- sample(c('HH', 'HT', 'TT'), size = n, replace = TRUE,
                 prob = c(0.36, 0.48, 0.16)) # P(H) = 0.6
table(tosses)
```

```
## tosses
## HH HT TT
## 35 46 19
```

```
sum((c(35, 46, 19) - n*c(0.25, 0.5, 0.25))^2 /
        n*c(0.25, 0.5, 0.25))
```

```
## [1] 0.42
```

```
pchisq(0.42, df = 1, lower.tail = FALSE)
```

```
## [1] 0.516937
```

Aside on the χ^2 Test: Example

```
set.seed(72364)
n <- 1000
tosses <- sample(c('HH', 'HT', 'TT'), size = n, replace = TRUE,
                 prob = c(0.36, 0.48, 0.16)) # P(H) = 0.6
table(tosses)
```

```
## tosses
##  HH  HT  TT
## 384 459 157
```

```
sum((c(384, 459, 157) - n*c(0.25, 0.5, 0.25))^2 /
        n*c(0.25, 0.5, 0.25))
```

```
## [1] 7.49175
```

```
pchisq(7.492, df = 1, lower.tail = FALSE)
```

```
## [1] 0.006197369
```

Asside on the χ^2 Test: Applications

- Potential uses:
  - Test HWE
  - Check HLA-A allele distribution in a sample, compared to known reference
- Not appropriate for:
  - Test mean blood pressure (not categorical)
  - Test HLA-A and HLA-B allele distributions (frequencies must add to 1; would need two separate tests).

Allele Flips: Problem

It is possible that you will get major alleles that are reported on different strands. In this example, we have an allele flip between sample A and sample B, which results in very different allele frequencies (reference is T).

- Sample A: $f_T = 0.23$
- Sample B: $f_T = 0.81$

If we flip sample B to the other strand we get an allele frequency of 0.19, which is much more in line with what we would expect.

Figure 9: Picking different strands as the reference in different samples is called an allele flip

Allele Flips: Solution

If you aren't merging multiple samples, this shouldn't be a problem. Also, if you don't identify any allele flips in A/T or C/G SNPs while merging multiple samples, there probably aren't any allele flips in A/T or C/G SNPs (this isn't a guaruntee, though).

- If neither of those is true, you can check for allele flips using the `--flip-scan` module in PLINK.
- This works by scanning all potential allele flips (i.e. A/T and C/G SNPs) for inconsistent linkage disequilibrium between cases and controls. Thus, you may need to create a dummy case/control variable that correspnods to the sample ID for this test.

```
plink --bfile mydata --flip-scan
```

Allele Flips: Solution

```
## CHR     Chromosome
## SNP     SNP identifier for index SNP
## BP      Base-pair position
## A1      Minor allele code
## A2      Major allele code
## F       Allele frequency (A1 allele)
## POS     Number of positive LD matches
## R_POS   Average correlation of these
## NEG     Number of negative LD matches
## R_NEG   Average correlation of these
## NEGSNPS The SNPs showing negative correlation
```

CHR	SNP	BP	A1	A2	F	POS	R_POS	NEG	R_NEG	NEGSNP
14	rs12434442	72158039	T	C	0.249	5	0.515	1	0.46	rs2240
14	rs4899437	72190986	G	C	0.394	5	0.802	1	0.987	rs2240
14	rs2803980	72196284	G	A	0.41	5	0.808	1	0.95	rs2240
14	rs2240344	72197893	C	G	0.489	0	NA	7	0.807	rs1243
14	rs2286068	72198107	C	T	0.407	7	0.741	1	0.962	rs2240
14	rs7160830	72209491	T	C	0.414	6	0.801	1	0.922	rs2240
14	rs10129954	72220454	T	C	0.413	6	0.729	1	0.73	rs2240
14	rs7140455	72240734	T	C	0.469	4	0.72	1	0.64	rs2240

Allele Flips: Solution

Any SNPs you identify that need to be flipped can be recoded as follows:

```
plink --bfile mydata --flip list.txt --recode
```

If your samples have already been merged, you can do this for just a subset of the data:

```
plink --bfile mydata --flip list.txt
      --flip-subset subsample.txt --recode
```

Sex Mismatch

The reported sex of an individual may not match their biological sex. We want to be aware of these mismatches and treat them accordingly in our analysis. These can be checked with the following command:

```
plink --bfile mydata --sex-check
```

This will generate a file called `plink.sexcheck`.

```
## FID     Family ID
## IID     Individual ID
## PEDSEX  Sex as determined in pedigree file (1=male, 2=female)
## SNPSEX  Sex as determined by X chromosome STATUS  Displays "PROBLEM" or
##         "OK" for each individual
## F       The actual X chromosome inbreeding (homozygosity) estimate
```

Relatives: Identification

We also want to know if there are relatives in the sample. An assumption for nearly any statistical test is that the sample is made up of independent individuals.

```
plink --file mydata --genome
```

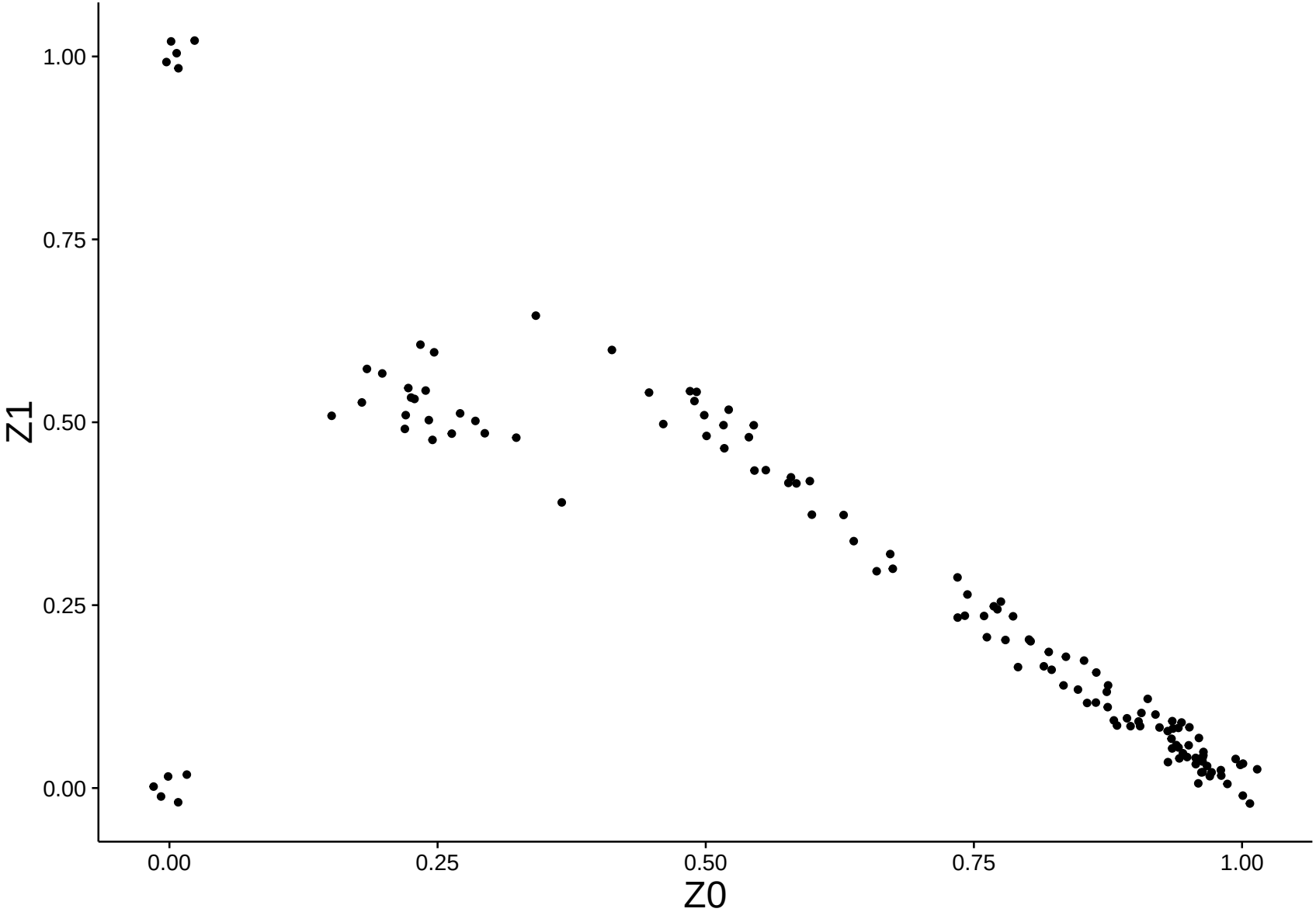
```
## FID1      Family ID for first individual
## IID1      Individual ID for first individual
## FID2      Family ID for second individual
## IID2      Individual ID for second individual
## RT        Relationship type given PED file
## EZ        Expected IBD sharing given PED file
## Z0        P(IBD=0)
## Z1        P(IBD=1)
## Z2        P(IBD=2)
## PI_HAT    P(IBD=2)+0.5*P(IBD=1) ( proportion IBD )
## PHE       Pairwise phenotypic code (1,0,-1 = AA, AU and UU pairs)
## DST       IBS distance (IBS2 + 0.5*IBS1) / ( N SNP pairs )
## PPC       IBS binomial test
## RATIO     Of HETHET : IBS 0 SNPs (expected value is 2)
```

Relatives: Z0, Z1, Z2

Relationship	Z0	Z1	Z2
Twins	0	0	1
Parent-Child	0	1	0
Full Sib	0.25	0.5	0.25
Half Sib	0.5	0.5	0
1st Cousin	0.75	0.25	0
2nd Cousin	0.875	0.125	0
nth Cousin	$1 - Z1$	0.5^{n+1}	0

Table 1: Probability of sharing 0, 1, or 2 loci Identically by Descent (Z0, Z1, and Z2, respectively)

Relatives: IBD Plot



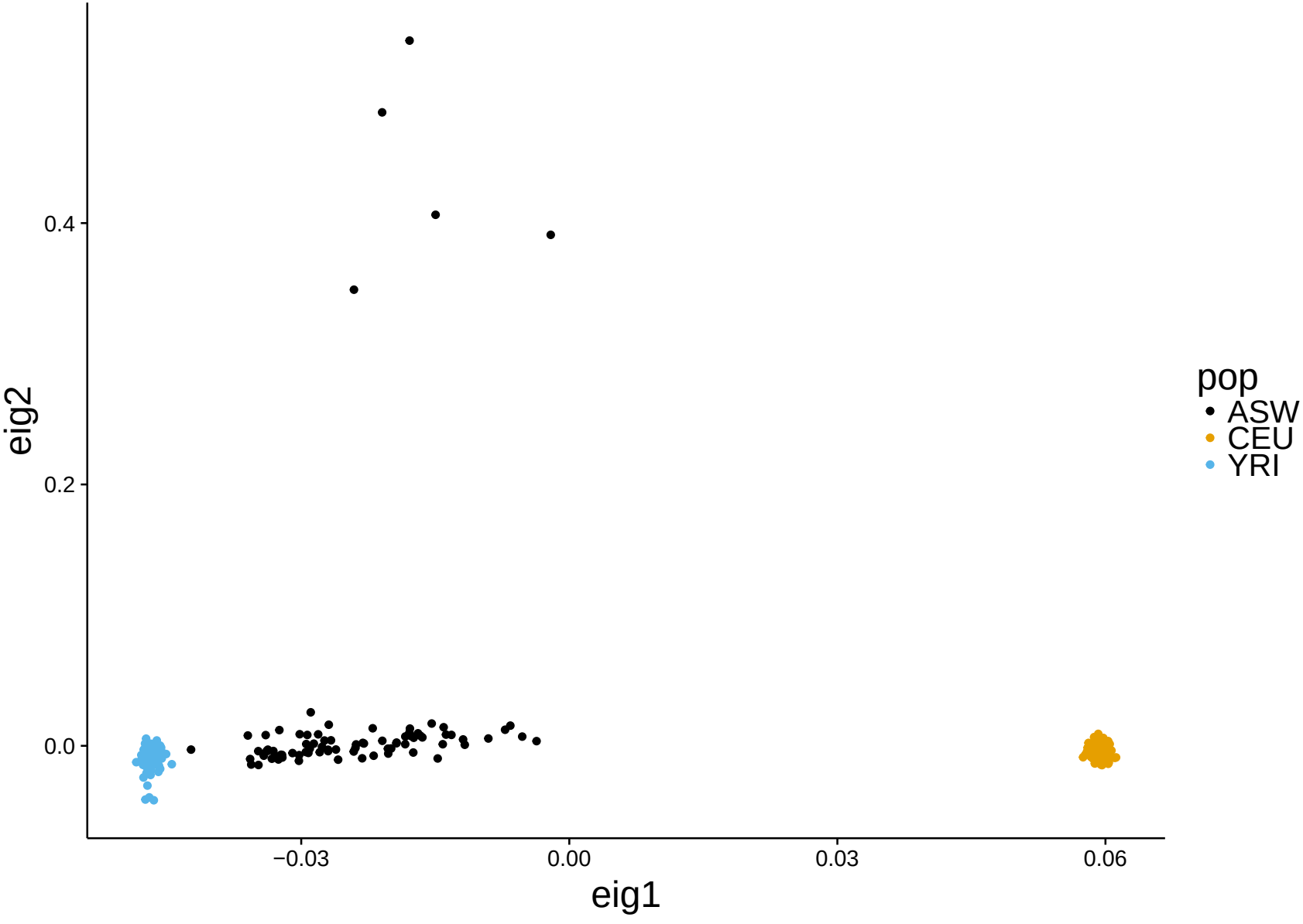
Population Structure

```
# start by merging files
snpgdsCombineGeno(paste0('~/Documents/HapMap3/gds/',
                         c('ASW', 'CEU', 'YRI'), '.gds'),
                  '~/Documents/HapMap3/gds/popStruct.gds',
                  name.prefix = c('ASW', 'CEU', 'YRI'))

# open new file
all <- snpgdsOpen('~/Documents/HapMap3/gds/popStruct.gds')

# calculate eigenvectors
structr <- snpgdsPCA(all)
save(structr, file = '../Data/13_structr.RData')

# close file
snpgdsClose(all)
```

Population Structure

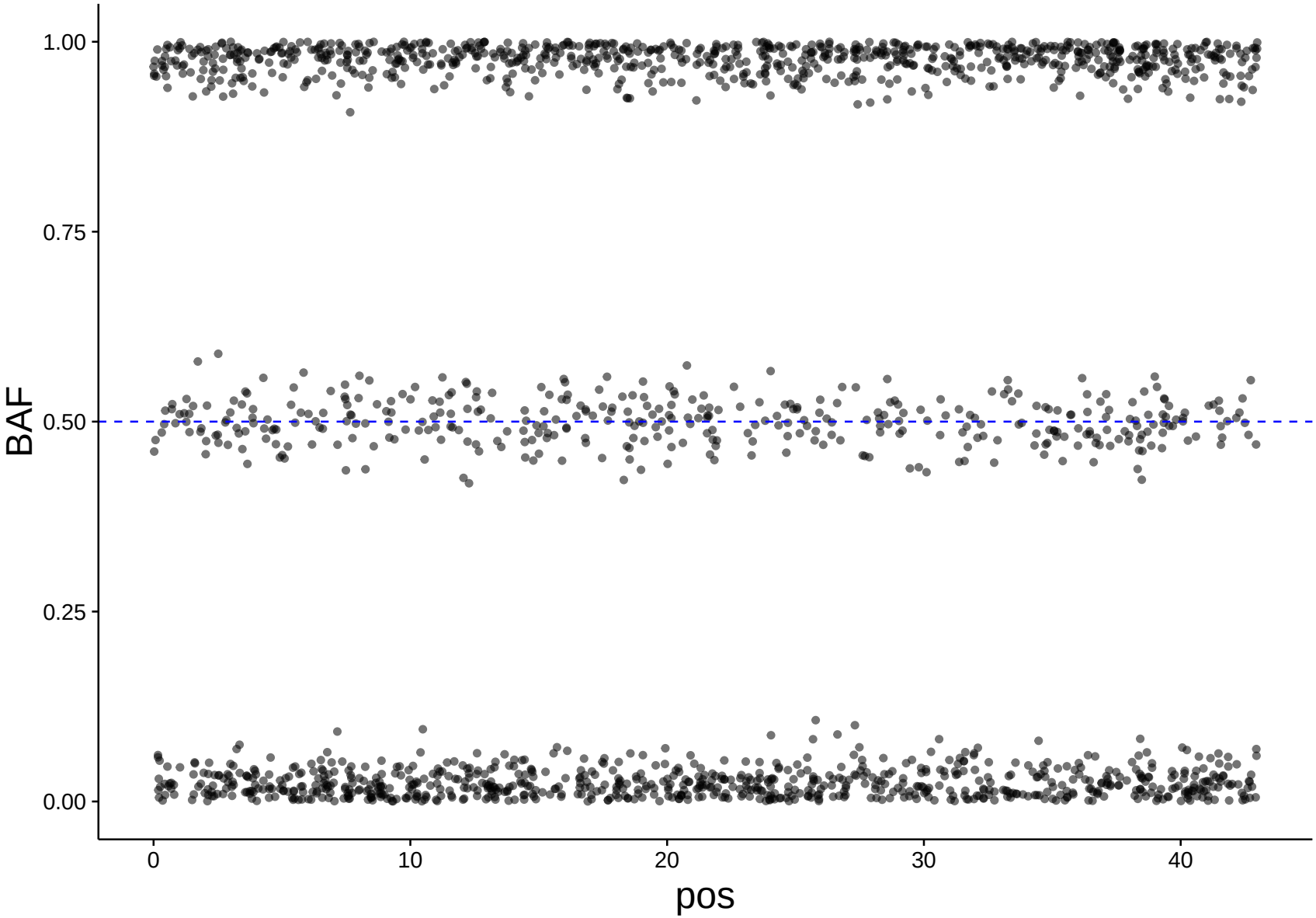
```
load('../Data/13_structr.RData')

g <- data_frame(eig1 = structr$eigenvect[,1],
                eig2 = structr$eigenvect[,2],
                pop = {strsplit(structr$sample.id, '.',
                                fixed = TRUE) %>%
                       sapply(`[`, 1)}) %>%
    ggplot(aes(eig1, eig2, color = pop)) +
    geom_point() +
    scale_color_manual(values = cbbPalette)
```

Population Structure



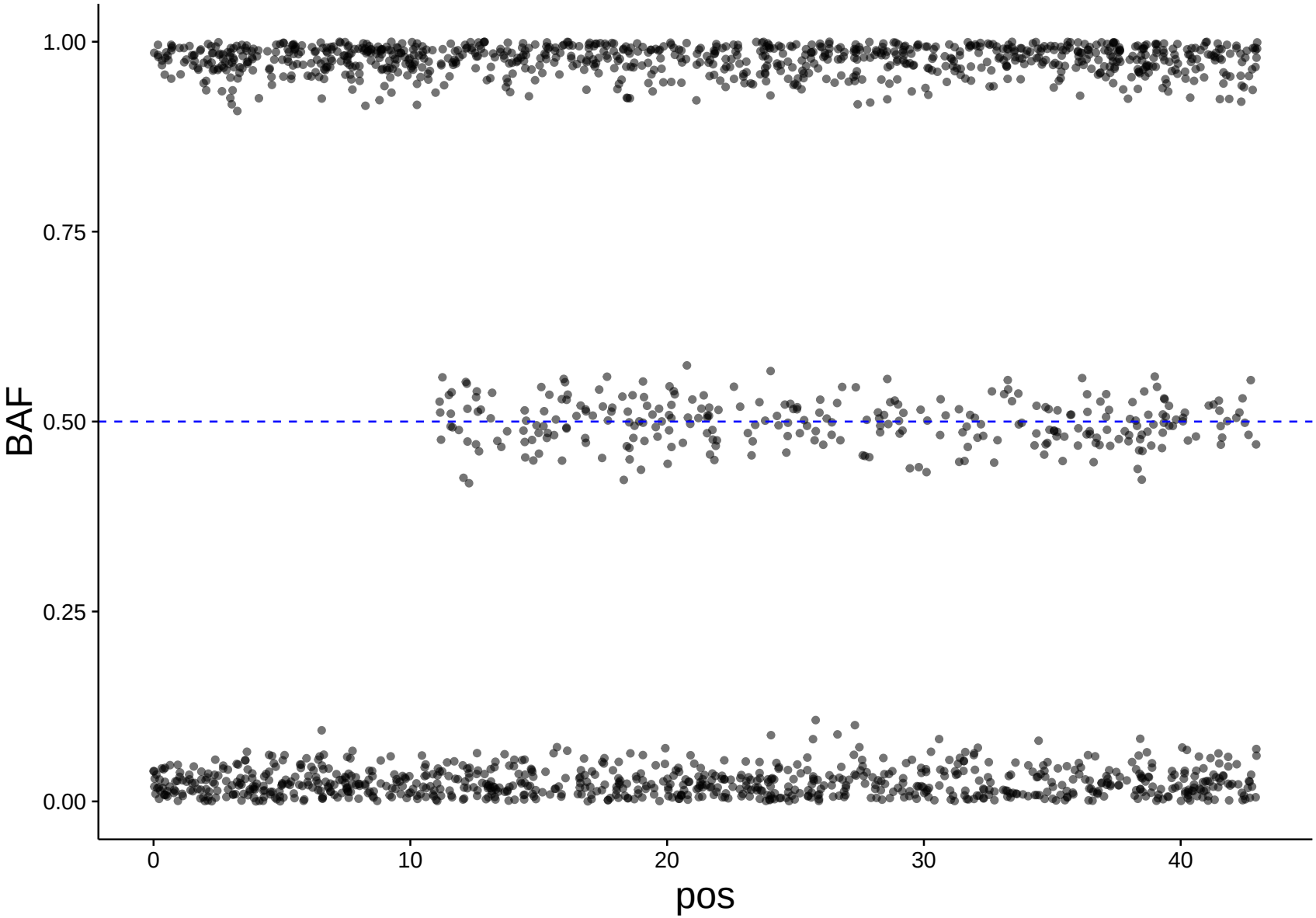
B Allele Plots

A normal BAF plot:

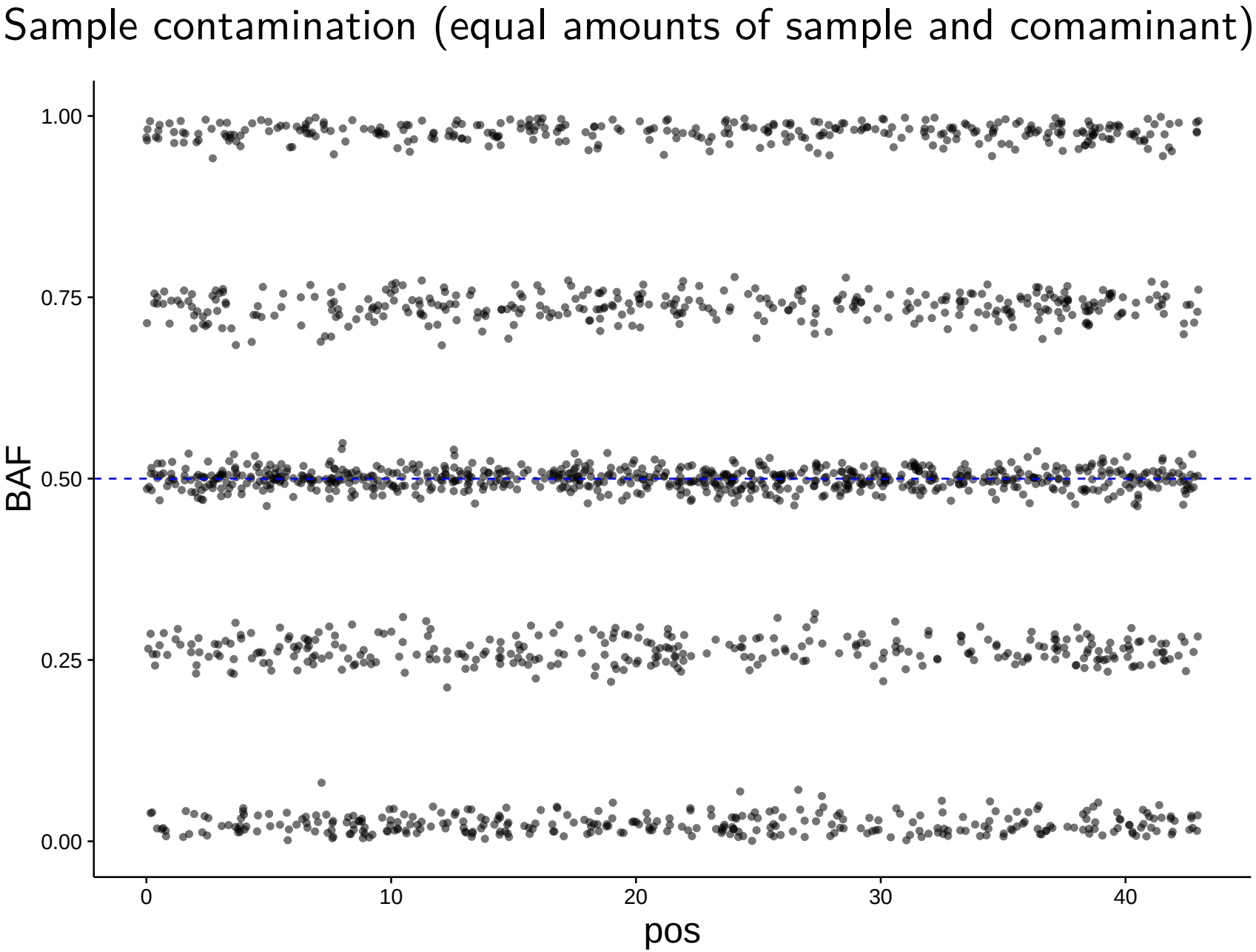


B Allele Plots

Large deletion leading to a loss of heterozygosity:

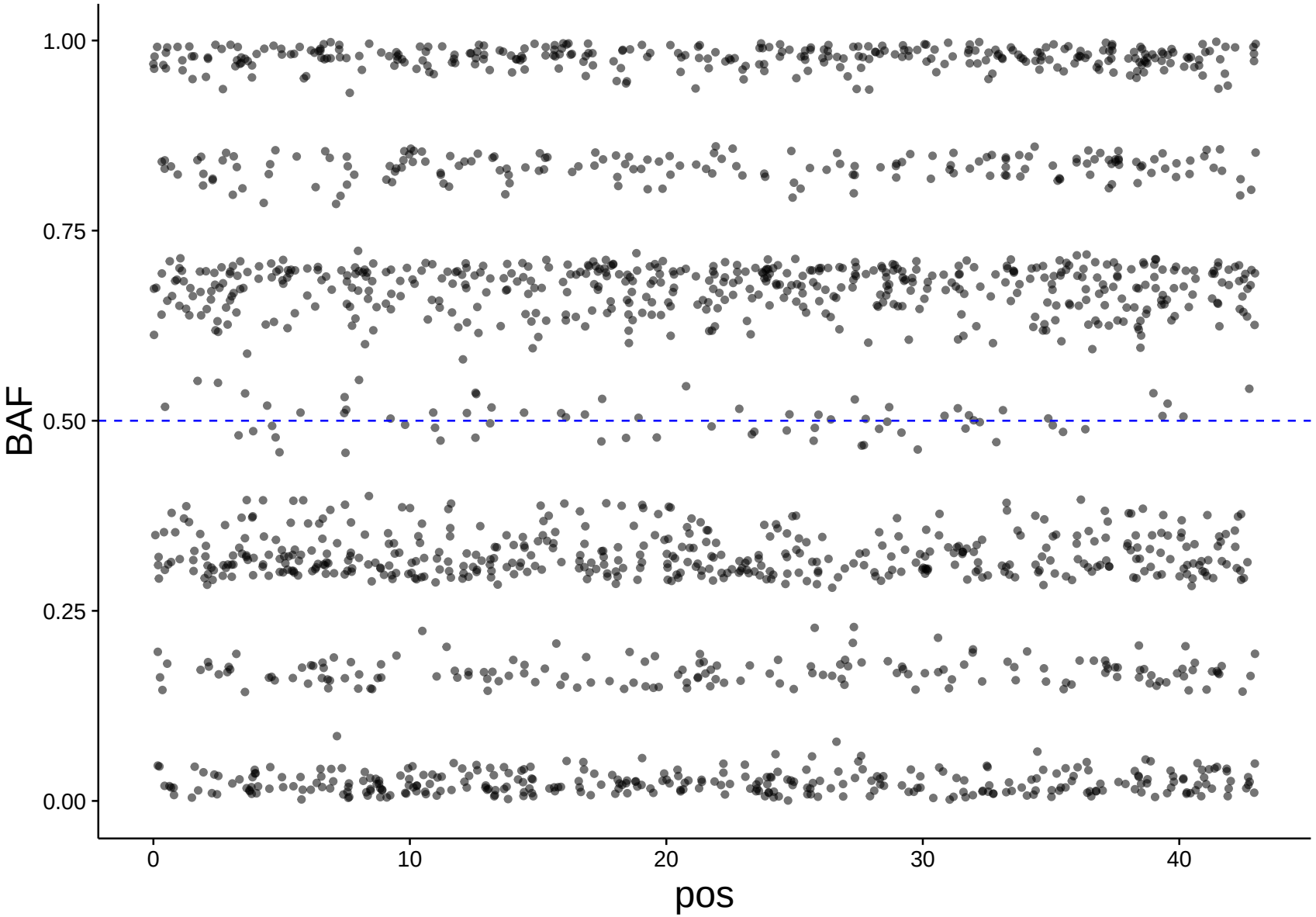


B Allele Plots



B Allele Plots

Sample contamination (30% of DNA is contamination)



B Allele Plots

Sample contamination (10% of DNA is contamination)