PGRNseq Sample and Phenotype Processing

```
pheno.df <- data.frame(id=nomiss.pheno$SM_ID,
                       sex=as.factor(nomiss.pheno$Sex),
                       ANC=as.numeric(nomiss.pheno$ANC_nadir_1000cells.per.mcL),
                       site=as.factor(nomiss.pheno$Study.site),
                       race=as.factor(nomiss.pheno$Race),
                       dose=as.factor(nomiss.pheno$Dose..mg.))
```

Data

Sample size:

`nrow(pheno.df)` samples in the phenotype file. `sum(!is.na(pheno.df$ANC))` samples that have a non-missing, non-zero Neutrophil count phenotype.

Phenotypes:

Neutrophil Count Nadir (1000 cells/mcL) log Neutrophil Count Nadir Inverse normalized Neutrophil Count Nadir.

Covariates:

Sex :: factor Site of ascertainment :: factor Dosage regimen :: continuous

observed Distributions of Phenotypes

```
anc.plt <- ggplot(pheno.df, aes(x=ANC)) + geom_histogram()
print(anc.plt + ggtitle('ANC'))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

```
loganc.plt <- ggplot(pheno.df, aes(x=log(ANC))) + geom_histogram()
print(loganc.plt + ggtitle('log(ANC)'))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

```
loganc.plt <- ggplot(pheno.df, aes(x=my.invnorm(ANC))) + geom_histogram()
print(loganc.plt + ggtitle('Inverse Normalized ANC'))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

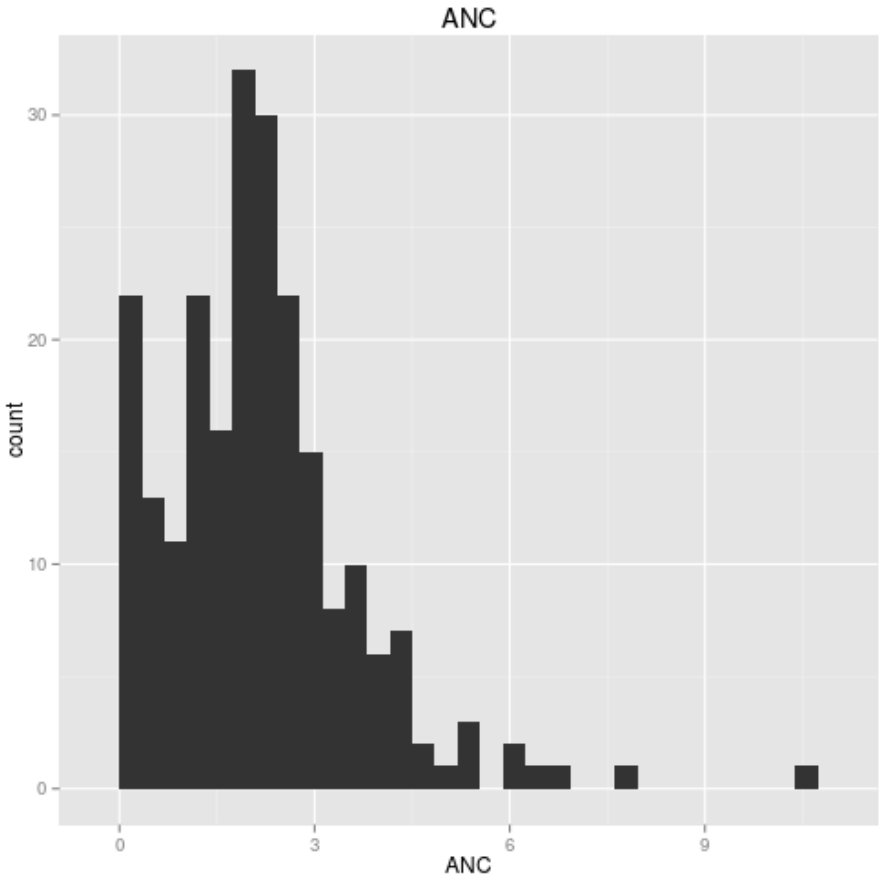


Figure 1: Distribution of the Adjusted Neutrophil Count (ANC).

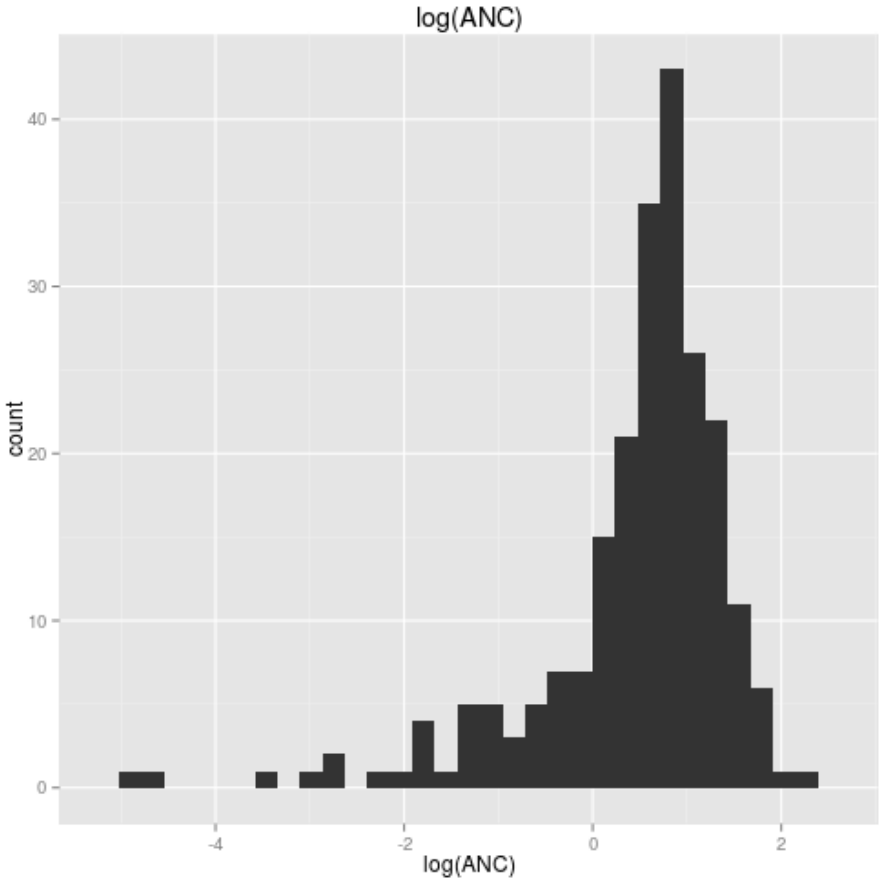


Figure 2: Distribution of the log transformation of the the Adjusted Neutrophil Count (ANC).

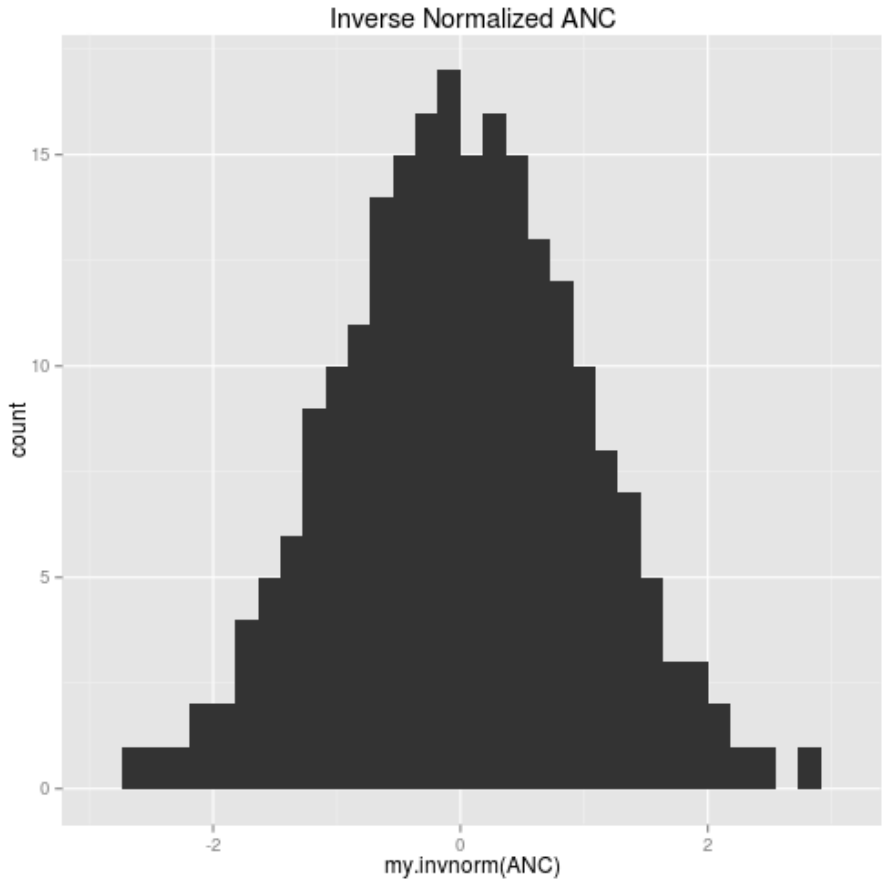


Figure 3: Distribution of the inverse normalized Adjusted Neutrophil Count (ANC). Inverse normalization consists of ranking the samples by their phenotype, and mapping those ranks to values in [0, 1]. These values are then fed through the standard normal quantile function, and the result is a normal distribution with samples mainting their relative order based on their original phenotype. The downside is that any subsequent regression loses biological interpretability.

Exome chip:

PCA

Prune SNP set for LD, filter on MAF, and run PCA.

```
ld.snpset <- snpgdsLDpruning(exome.geno)
```

```
## SNP pruning based on LD:
##  Sliding window: 500000 basepairs, Inf SNPs
##  |LD| threshold: 0.2
## Removing 5465 non-autosomal SNPs
## Removing 154888 SNPs (monomorphic, < MAF, or > missing rate)
## Working space: 253 samples, 82543 SNPs
## Chromosome 1: 18.93%, 4671/24674
## Chromosome 2: 20.56%, 3535/17190
## Chromosome 3: 19.84%, 2883/14528
## Chromosome 4: 23.01%, 2347/10198
## Chromosome 5: 21.56%, 2439/11314
## Chromosome 6: 17.86%, 2710/15171
## Chromosome 7: 21.43%, 2377/11092
## Chromosome 8: 21.54%, 1922/8921
## Chromosome 9: 20.32%, 2073/10201
## Chromosome 10: 21.77%, 2071/9511
## Chromosome 11: 18.61%, 2890/15528
## Chromosome 12: 19.94%, 2456/12318
## Chromosome 13: 25.05%, 1097/4380
## Chromosome 14: 20.15%, 1556/7721
## Chromosome 15: 19.46%, 1610/8273
## Chromosome 16: 17.86%, 1855/10384
## Chromosome 17: 16.84%, 2183/12963
## Chromosome 18: 24.09%, 903/3749
## Chromosome 19: 15.39%, 2302/14955
## Chromosome 20: 19.28%, 1239/6428
## Chromosome 21: 20.76%, 586/2823
## Chromosome 22: 18.50%, 945/5109
## 46650 SNPs are selected in total.
```

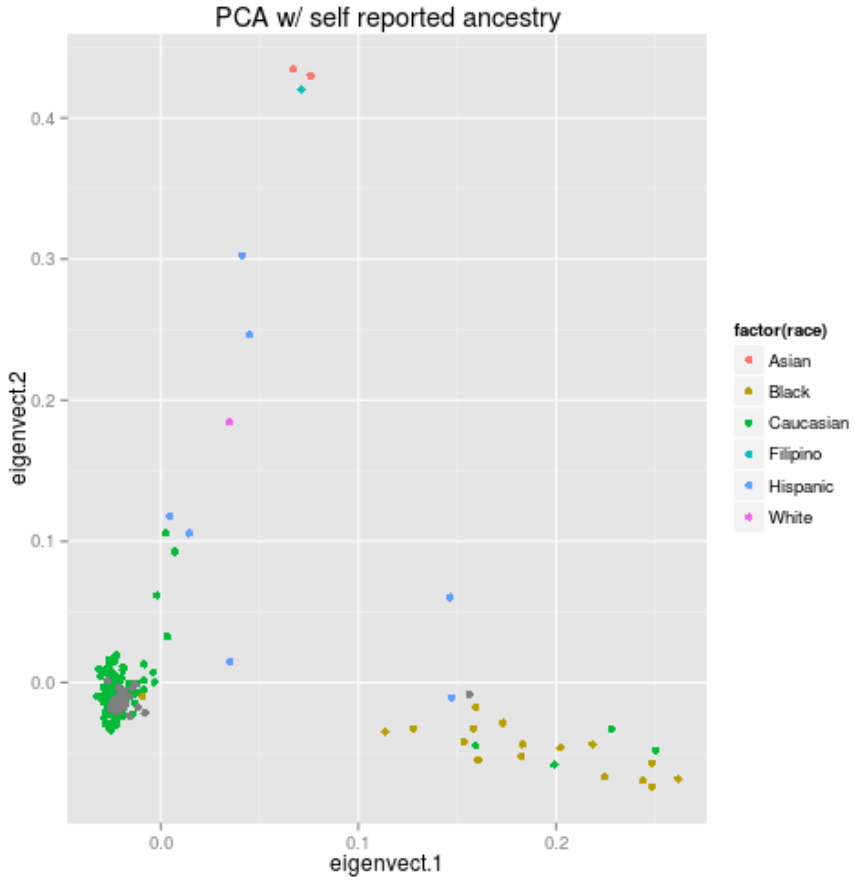
```
pca <- snpgdsPCA(exome.geno, snp.id = unlist(ld.snpset), maf = 0.05)
```

```
## Principal Component Analysis (PCA) on SNP genotypes:
## Removing 35315 SNPs (monomorphic, < MAF, or > missing rate)
## Working space: 253 samples, 11335 SNPs
##  Using 1 CPU core.
```

```
## PCA: the sum of all working genotypes (0, 1 and 2) = 1646004
## PCA: Tue Aug 26 16:04:57 2014    0%
## PCA: Tue Aug 26 16:04:57 2014    100%
## PCA: Tue Aug 26 16:04:57 2014    Begin (eigenvalues and eigenvectors)
## PCA: Tue Aug 26 16:04:57 2014    End (eigenvalues and eigenvectors)
```

```
pca.df <- data.frame(sample.id=pca$sample.id, eigenvect=pca$eigenvect)
pheno.df <- merge(pca.df, pheno.df, by.x = "sample.id", by.y = "id")
```

Look at the first two PCs, and color the samples by self-reported ancestry.

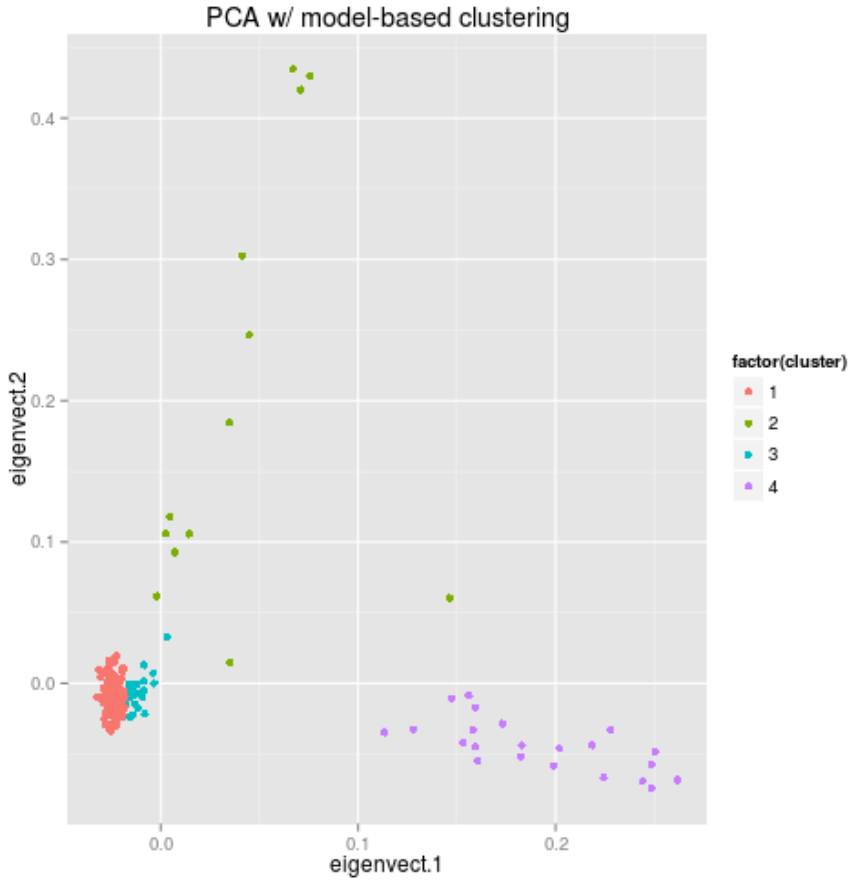


Model-based Clustering

Add principal components to the phenotype data frame, then cluster samples in the first two PCs.

```
names(pheno.df)[which(names(pheno.df) == "sample.id")] <- "id"
## specify 3 mixture components
cluster3 <- Mclust(pheno.df[,c("eigenvect.1", "eigenvect.2")], G=3)
## select # of components using BIC
clustern <- Mclust(pheno.df[,c("eigenvect.1", "eigenvect.2")])
pheno.df$cluster <- clustern$classification
## euro clust is largest
euroclust <- which(table(pheno.df$cluster) == max(table(pheno.df$cluster)))
pheno.df$iseuro <- pheno.df$cluster == euroclust
```

Examine the results of our PCA and clustering the first two PCs.



Taking a closer look at the clustering of the Europeans. We may be able to explain some genetic variation just by site of ascertainment.

```
## Warning: Removed 35 rows containing missing values (geom_point).
```

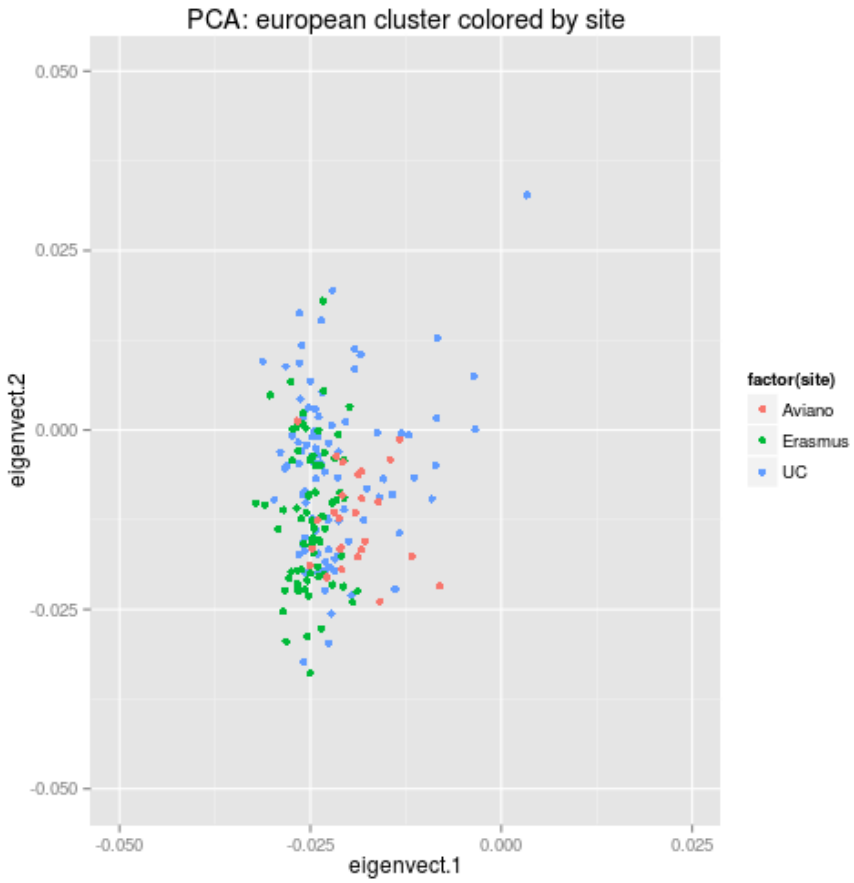


Figure 4: plot of chunk euro_zoom_plot