Practical Batch-Effects

Maarten van Iterson

July 4, 2017

Contents

Introduction

In this practical you will learn how to perform batch correction on RNAseq data using the packages RUVSeq and sva. The example data that will be used is a subset of the HapMap RNAseq data described by Pickrell and Montgomery (Pickrell et al. 2010, Montgomery et al. (2010)). A combined dataset containing the RNAseq data of both papers is available from the ReCount website (Frazee, Langmead, and Leek 2011). As phenotype information we have Population (CEU/YRI) and Gender (Male/Female). In some of the Exercises we will assume the population origin of the samples is a unknown batch effect.

The practical consists of four parts:

- download the data and some preprocessing
- find diff. expr. genes between Male/Female using limma's `voom` or edgeR with population as a known batch in the design matrix
- estimate batch effects using RUVSeq and find diff. expr. genes
- estimate batch effects using sva and find diff. expr. genes

Get the data:

Run the following two code chunks to get the data in your R-environment.

```
library(Biobase)
monpick <- "http://bowtie-bio.sourceforge.net/recount/ExpressionSets/montpick_eset.RData"
load(url(monpick))
head(pData(montpick.eset))
```

```
##         sample.id num.tech.reps population      study
## NA06985   NA06985             1        CEU Montgomery
## NA06986   NA06986             1        CEU Montgomery
## NA06994   NA06994             1        CEU Montgomery
## NA07000   NA07000             1        CEU Montgomery
## NA07037   NA07037             1        CEU Montgomery
## NA07051   NA07051             1        CEU Montgomery
```

```
counts <- exprs(montpick.eset)
counts[1:5, 1:5]
```

```
##                 NA06985 NA06986 NA06994 NA07000 NA07037
## ENSG00000000003       0       0       0       1       0
## ENSG00000000005       0       0       0       0       0
## ENSG00000000419      11      16      10      19       8
## ENSG00000000457      28      22      17      21      18
## ENSG00000000460       0       7       2       4       6
```

Unfortunately, gender information is not provided. This can be obtained from the 1000genomes website: ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606_sample_info/20130606_sample_info.xlsx Download the sample information file and save the sheet with sample info as csv-file and preprocess like this:

```
sample.info.file <- "https://raw.githubusercontent.com/molepi/MOLMED2017/master/Practical_BatchEffect/2
sample.info <- read.table(url(sample.info.file), header=TRUE, sep="\t")
head(sample.info)
```

```
##    Sample Family.ID Population          Population.Description Gender
## 1 HG00096   HG00096        GBR British in England and Scotland   male
## 2 HG00097   HG00097        GBR British in England and Scotland female
## 3 HG00098   HG00098        GBR British in England and Scotland   male
## 4 HG00099   HG00099        GBR British in England and Scotland female
## 5 HG00100   HG00100        GBR British in England and Scotland female
## 6 HG00101   HG00101        GBR British in England and Scotland   male
##   Relationship Unexpected.Parent.Child Non.Paternity Siblings Grandparents
## 1
## 2
## 3
## 4
## 5
## 6
##   Avuncular Half.Siblings Unknown.Second.Order Third.Order Other.Comments
## 1                                                                     NA
## 2                                                                     NA
## 3                                                                     NA
## 4                                                                     NA
## 5                                                                     NA
## 6                                                                     NA
```

```
pdata <- merge(pData(montpick.eset), sample.info, by.x="sample.id", by.y="Sample", all.x=TRUE)
pdata[is.na(pdata$Gender),]
```

```
##     sample.id num.tech.reps population    study Family.ID Population
## 117   NA19192             2        YRI Pickrell      <NA>       <NA>
## 118   NA19193             2        YRI Pickrell      <NA>       <NA>
##     Population.Description Gender Relationship Unexpected.Parent.Child
## 117                   <NA>   <NA>         <NA>                    <NA>
## 118                   <NA>   <NA>         <NA>                    <NA>
##     Non.Paternity Siblings Grandparents Avuncular Half.Siblings
## 117          <NA>     <NA>         <NA>      <NA>          <NA>
## 118          <NA>     <NA>         <NA>      <NA>          <NA>
##     Unknown.Second.Order Third.Order Other.Comments
## 117                 <NA>        <NA>             NA
## 118                 <NA>        <NA>             NA
```

```
counts <- counts[,!is.na(pdata$Gender)]
pdata <- pdata[!is.na(pdata$Gender),]
pdata <- droplevels(pdata)
dim(pdata)
```

```
## [1] 127  18
```

```
dim(counts)
```

```
## [1] 52580   127
```

We had to remove two samples because for these no phenotype information was available. Furthermore, we dropped the additional factor levels as these will interfere later with the creation of design-matrices.

Now we can start our analyzes!

Finding diff. expr. genes using `limma` or `edgeR`

First we will find diff. expr. genes using either limma's `voom` or edgeR which one is up to you!

Preprocessing and data inspection

> Exercise 1: Construct a *egdeR* `DGEList` with a group-variable the Gender information. Optionally, remove low expressed genes.

Solution 1:

```
isexpr <- rowSums(counts) > 50
counts <- counts[isexpr,]
dim(counts)
```

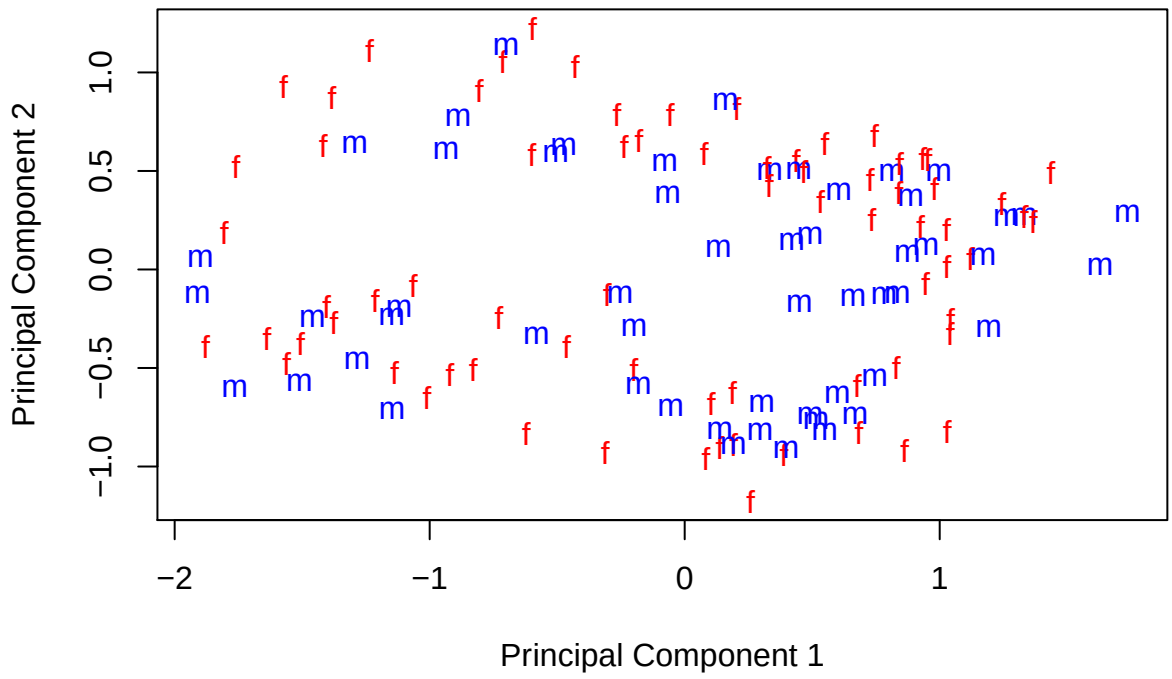
```
## [1] 9085  127
```

```
library(edgeR)
d <- DGEList(counts, group=pdata$Gender)
d <- calcNormFactors(d)
```

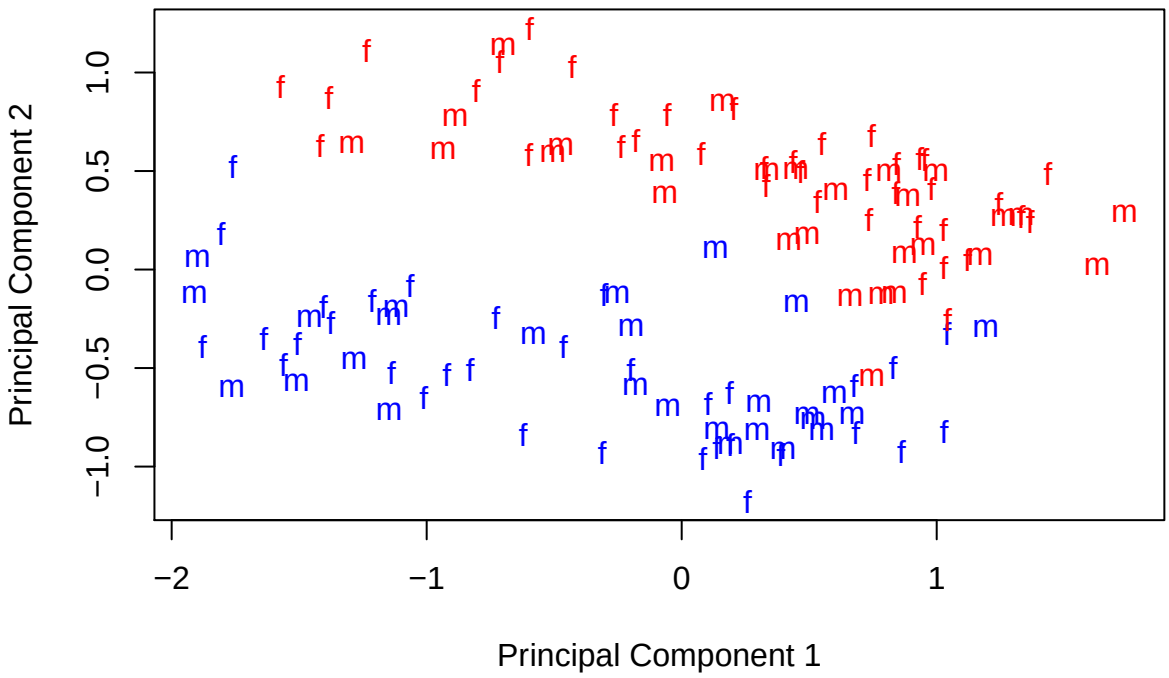
Exercise 2: Inspect the data using a Multi-dimensional scaling plot to see potential batches in the data. Use coloring and labels to see which, gender or population, has the strongest effect.

Solution 2:

```
Gender <- substring(pdata$Gender,1,1)
colGen <- ifelse(Gender=="m","blue","red")
plotMDS(d, labels = Gender,top = 50, col=colGen, gene.selection="common", prior.count = 5)
```




```
Population <- substring(pdata$population,1,1)
colPop <- ifelse(Population=="C","blue","red")
plotMDS(d, labels = Gender,top = 50, col=colPop, gene.selection="common", prior.count = 5)
```

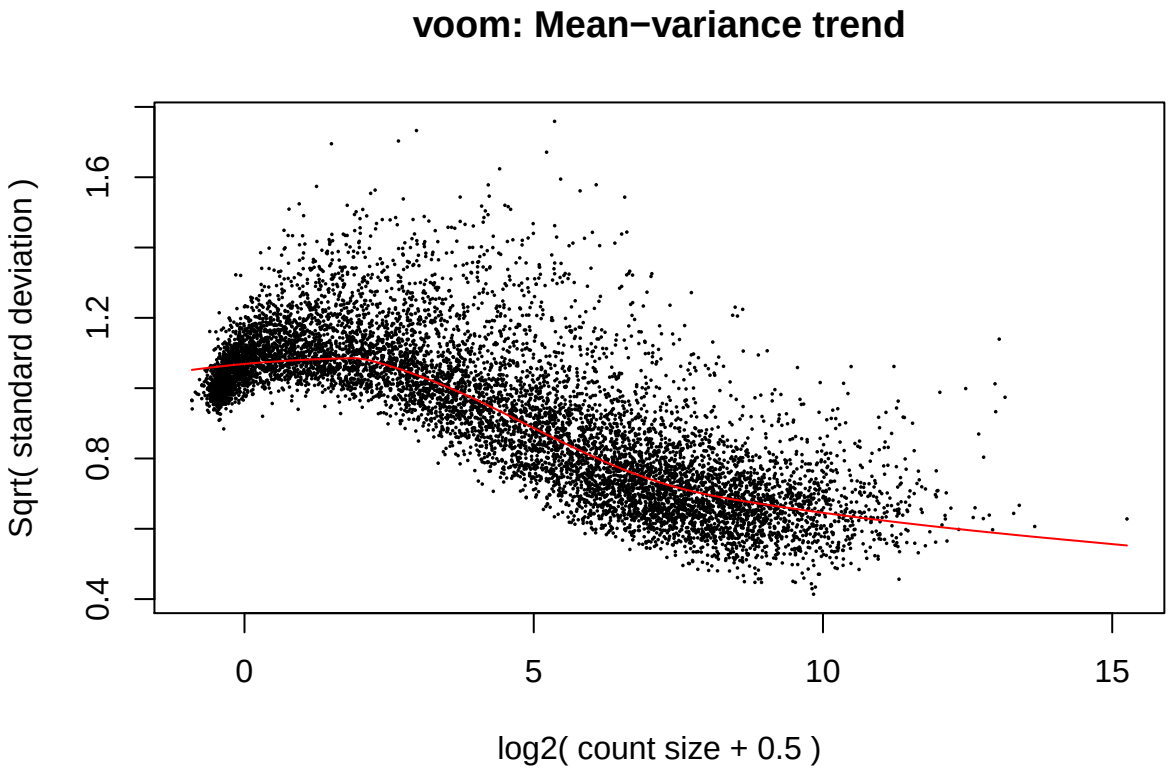


Fitting using `voom`

Exercise 3a: Fit a linear model correcting for the population structure using `voom`.

Solution 3a:

```
design <- model.matrix(~Gender + Population, data = pdata)
v <- voom(d, design, plot = TRUE)
```




```
fit <- lmFit(v, design)
fit <- eBayes(fit)
```

```
summary(decideTests(fit))
```

```
##    (Intercept) Gendermale PopulationYRI
## -1        1490          8          2505
## 0          720       9059          3780
## 1         6875         18          2800
```

```
topTable(fit)
```

```
## Removing intercept from test coefficients
```

```
##                 Gendermale PopulationYRI     AveExpr         F
## ENSG00000129824  8.59843720   -0.10666014 2.303870943 1331.0881
## ENSG00000099749  5.38177831    0.01755708 0.560818136  548.0786
## ENSG00000154620  4.76975249   -0.76318939 0.917947739  459.6110
## ENSG00000198692  4.34208054   -0.07781427 0.062664870  252.1678
## ENSG00000157828  3.88044378   -0.79429515 0.003851596  211.1734
## ENSG00000175265  0.34974300    4.13948895 2.492853178  155.2317
## ENSG00000172638 -0.03352651    3.73612155 3.513702544  147.6890
## ENSG00000152795  0.15797888    2.94887391 0.980299050  138.5802
## ENSG00000135473  0.26366058    1.77986395 7.215823299  133.3356
```

```
## ENSG00000033030  0.11170019    1.04261373 6.834586594  128.3276
##                       P.Value    adj.P.Val
## ENSG00000129824 4.848864e-88 4.405193e-84
## ENSG00000099749 1.521134e-64 6.909753e-61
## ENSG00000154620 4.258737e-60 1.289688e-56
## ENSG00000198692 9.587573e-46 2.177578e-42
## ENSG00000157828 8.437196e-42 1.533039e-38
## ENSG00000175265 2.364099e-35 3.579640e-32
## ENSG00000172638 2.322201e-34 3.013885e-31
## ENSG00000152795 4.092192e-33 4.647196e-30
## ENSG00000135473 2.263856e-32 2.285237e-29
## ENSG00000033030 1.209844e-31 1.099143e-28
```

Exercise 3b: Fit a linear model correcting for the population structure using *edgeR* (you can reuse the `DGEList`).

Solution 3b:

```
d1 <- estimateGLMCommonDisp(d, design)
d1 <- estimateGLMTagwiseDisp(d1, design)
```

```
fit <- glmFit(d1, design)
lrt <- glmLRT(fit, coef=2)
topTags(lrt)
```

```
## Coefficient:  Gendermale
##                       logFC    logCPM        LR        PValue           FDR
## ENSG00000099749  6.8774954 2.8201086 974.94616 5.013451e-214 4.554720e-210
## ENSG00000154620  4.9801496 2.8152210 780.65508 8.670821e-172 3.938720e-168
## ENSG00000198692  5.9873934 2.0780918 480.17783 1.954558e-106 5.919052e-103
## ENSG00000157828  5.5959279 1.8638669 379.20317  1.855429e-84  4.214143e-81
## ENSG00000129824  7.5369623 6.0814568 164.60110  1.118018e-37  2.031440e-34
## ENSG00000006757 -1.0960616 5.4385579 128.17417  1.028128e-29  1.556757e-26
## ENSG00000183878  3.5841318 0.4280894 116.01507  4.716991e-27  6.121980e-24
## ENSG00000130021 -0.7834851 2.7535048  41.90753  9.569335e-11  1.086718e-07
## ENSG00000086712 -0.6133738 5.1559998  38.97430  4.294224e-10  4.334780e-07
## ENSG00000236696  2.1822011 0.7176931  34.49963  4.263316e-09  3.873222e-06
```

Optional Exercise: Annotate the top genes using org.Hs.eg.db. Are these the genes you would had expected?

Batch effect correction using *RUVseq*

Now we will assume the population of the samples was unknown and investigate if we can correct for this using the method implemented in *RUVseq*. Since, we do not have negative controls we will use a set of empirical controls. Empirical controls are just the genes that do not show diff. expr. for the phenotype of interest.

Exercise 4: Find a set of empirical control genes.

Solution 4:

```
top <- topTags(lrt, n=Inf)$table
empirical <- rownames(d)[which(!(rownames(d) %in% rownames(top)[1:5000]))]
```

Exercise 5: Run `RUVg` and inspect the effect on the data using the plotRLE and/or plotPCA from *RUVSeq*.

Solution 5:

```
library(RUVSeq)
rownames(pdata) <- pdata$sample.id
set <- newSeqExpressionSet(d$counts,
                           phenoData = data.frame(Gender=as.character(pdata$Gender),
                                                  row.names=pdata$sample.id))
set
```

```
## SeqExpressionSet (storageMode: lockedEnvironment)
## assayData: 9085 features, 127 samples
##   element names: counts, normalizedCounts, offset
## protocolData: none
## phenoData
##   sampleNames: NA06985 NA06986 ... NA19257 (127 total)
##   varLabels: Gender
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation:
```

```
corrected <- RUVg(set, empirical, k=1)
```

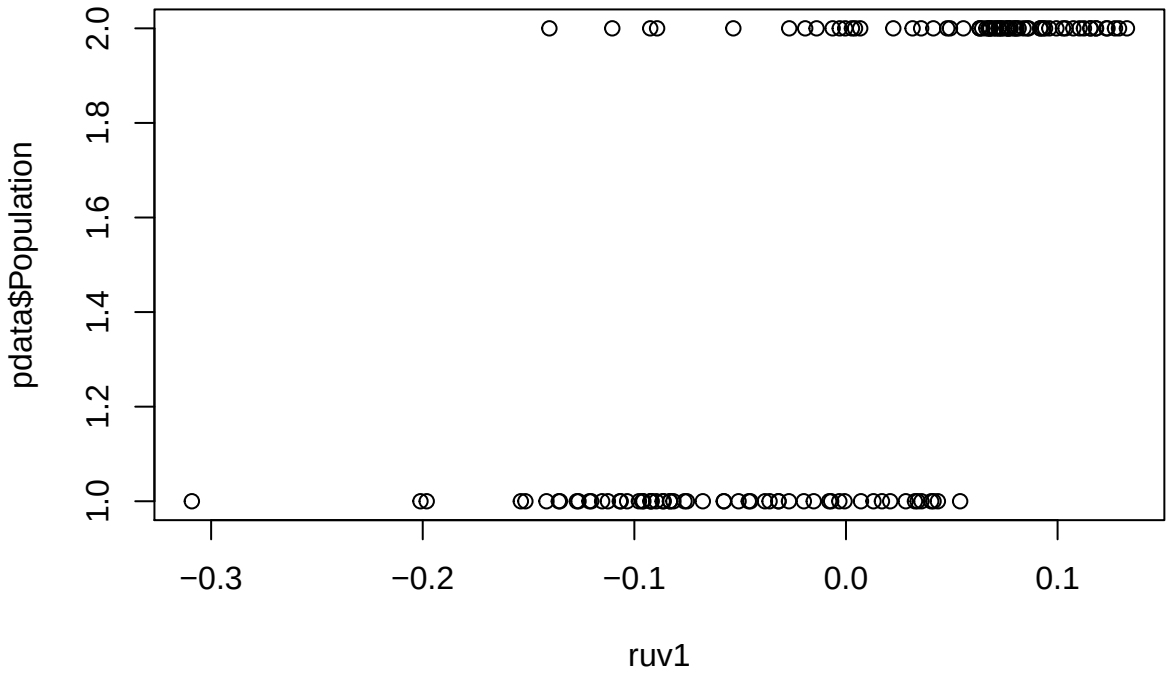
```
op <- par(mfcol=c(2, 1))
plotRLE(set, outline=FALSE, ylim=c(-2, 2), col=colPop, las=2)
plotRLE(corrected, outline=FALSE, ylim=c(-2, 2), col=colGen, las=2)
par(op)
```

```
op <- par(mfcol=c(2, 1))
plotPCA(set, col=colPop, cex=1.2)
plotPCA(corrected, col=colGen, cex=1.2)
par(op)
```

Exercise 6: Can you think of a way to see what the estimated unwanted variation represents? What does it represent?

Solution 6:

```
ruv1 <- corrected$W_1
plot(ruv1, pdata$Population)
```



Exercise 7: Fit a linear model correcting for the factor of unwanted variation. You can reuse the *DGEList* and again it is up to you to use `voom` or *edgeR*.

Solution 7:

```
designruv <- model.matrix(~Gender, data = pdata)
designruv <- cbind(designruv, ruv1)
d2 <- estimateGLMCommonDisp(d, designruv)
d2 <- estimateGLMTagwiseDisp(d2, designruv)
fit <- glmFit(d2, designruv)
lrt <- glmLRT(fit, coef=2)
topTags(lrt)
```

```
## Coefficient:  Gendermale
##                      logFC    logCPM        LR        PValue           FDR
## ENSG00000099749  6.8445763 2.8199557 895.19775 1.085865e-196 9.865079e-193
## ENSG00000154620  4.9844685 2.8159354 707.45369 7.159163e-156 3.252050e-152
## ENSG00000198692  5.9826299 2.0777222 478.77093 3.955369e-106 1.197817e-102
## ENSG00000157828  5.5428435 1.8652709 386.21667  5.514714e-86  1.252529e-82
## ENSG00000129824  8.4317350 6.0815047 260.19970  1.552338e-58  2.820597e-55
## ENSG00000006757 -1.0949151 5.4385432 129.21990  6.070615e-30  9.191923e-27
## ENSG00000183878  3.6192409 0.4281880 117.34140  2.416732e-27  3.136572e-24
## ENSG00000174938  2.7581269 1.0325940  42.96180  5.581908e-11  6.338955e-08
## ENSG00000086712 -0.6130102 5.1559425  38.76326  4.784482e-10  4.829668e-07
## ENSG00000236696  2.2462485 0.7178029  37.95131  7.253235e-10  6.589564e-07
```

Batch effect correction using *sva*

Now we will use the *sva*-package to estimate the unwanted variation introduced by the different populations.

> Exercise 8: What are the optimal number of surrogate variables to we should correct for?

Solution 8:

```
suppressPackageStartupMessages(library(sva))
designsva <- model.matrix(~Gender, data = pdata)
n.sv <- num.sv(d$counts, designsva, method="leek")
n.sv
```

```
## [1] 125
```

If find this number too high and we can not even correct for all these; you will get a error-message if you try. I suppose we use just one!

> Exercise 9: To estimate the surrogate variable we need to define our null hypothesis. What is our null hypothesis? Run `svaseq`.

Solution 9:

```
designsva0 <- model.matrix(~1, data = pdata)
svseq <- svaseq(d$counts, designsva, designsva0, n.sv=1)
```

```
## Number of significant surrogate variables is:  1
## Iteration (out of 5 ):1  2  3  4  5
```

> Exercise 10: Can you think of a way to see what the estimated unwanted variation represents? What does it represent?

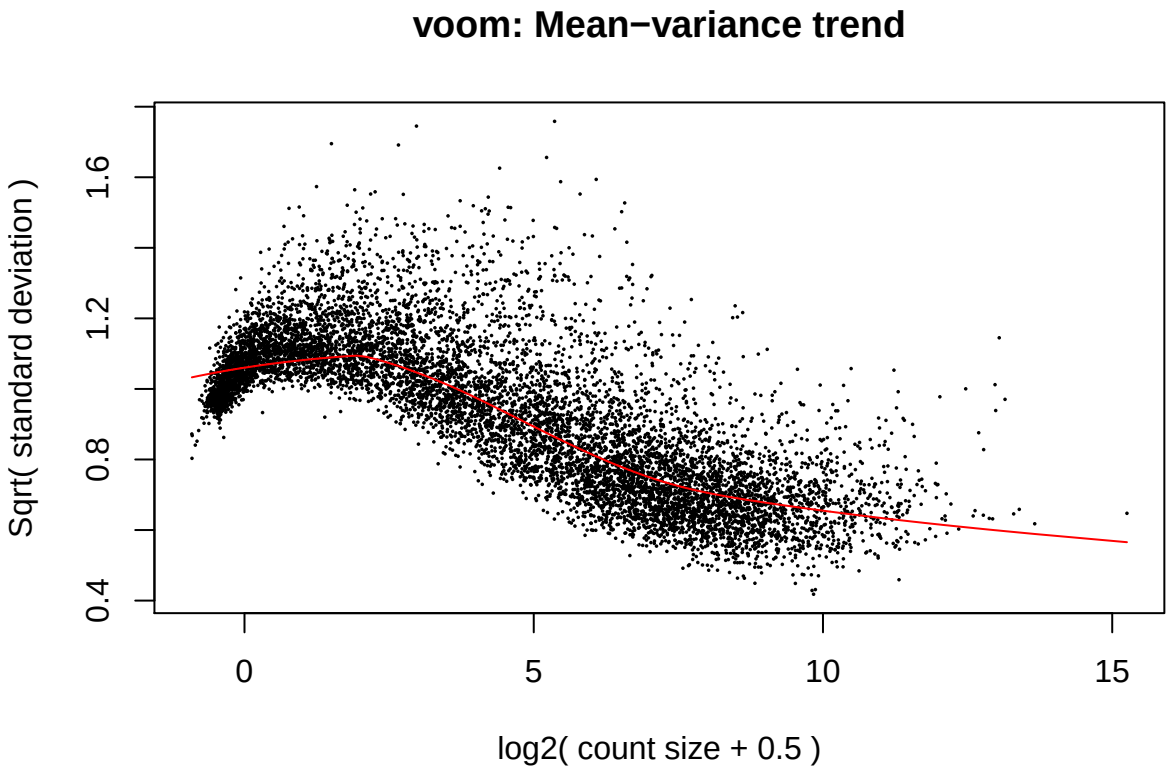
Solution 10:

```
plot(svseq$sv, pdata$Population)
```

> Exercise 11: Fit a linear model correcting for the surrogate variable. You can reuse the `DGEList` and again it is up to you to use `voom` or *edgeR*.

Solution 11:

```
designsva <- cbind(designsva, svseq$sv)
v <- voom(d, designsva, plot = TRUE)
```




```
fit <- lmFit(v, designsva)
fit <- eBayes(fit)
summary(decideTests(fit))
```

```
##    (Intercept) Gendermale
## -1        1555          7 2763
## 0          482       9053 3861
## 1         7048         25 2461
```

```
topTable(fit)
```

```
## Removing intercept from test coefficients
```

```
##                    Gendermale        V2      AveExpr          F
## ENSG00000129824  8.604150936  1.705214  2.303870943 1352.26926
## ENSG00000099749  5.391838113  2.565670  0.560818136  586.85774
## ENSG00000154620  4.802311088  3.740992  0.917947739  426.27746
## ENSG00000198692  4.339271289  2.025716  0.062664870  257.74453
## ENSG00000157828  3.923322619  6.244545  0.003851596  252.24194
## ENSG00000177324 -0.006480034  8.986664 -1.734992706   93.23657
## ENSG00000139329  0.090034950  9.414627 -1.677449269   87.06571
## ENSG00000183878  1.525424768  4.865944 -1.185817065   86.50908
## ENSG00000135473  0.228049100 -9.796073  7.215823299   83.46547
## ENSG00000103995  0.378832893 -8.829990  6.102712562   75.87484
```

```
##                      P.Value    adj.P.Val
## ENSG00000129824 1.746683e-88 1.586862e-84
## ENSG00000099749 2.647499e-66 1.202626e-62
## ENSG00000154620 3.127155e-58 9.470067e-55
## ENSG00000198692 3.018749e-46 6.856333e-43
## ENSG00000157828 9.334988e-46 1.696167e-42
## ENSG00000177324 6.006371e-26 9.094647e-23
## ENSG00000139329 8.122270e-25 1.054155e-21
## ENSG00000183878 1.032616e-24 1.172665e-21
## ENSG00000135473 3.898092e-24 3.934907e-21
## ENSG00000103995 1.209273e-22 1.098625e-19
```

References

Frazee, A. C., B. Langmead, and J. T. Leek. 2011. "ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets." *BMC Bioinformatics* 12: 449.

Montgomery, S. B., M. Sammeth, M. Gutierrez-Arcelus, R. P. Lach, C. Ingle, J. Nisbett, R. Guigo, and E. T. Dermitzakis. 2010. "Transcriptome genetics using second generation sequencing in a Caucasian population." *Nature* 464 (7289): 773–77.

Pickrell, J. K., J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nkadori, J. B. Veyrieras, M. Stephens, Y. Gilad, and J. K. Pritchard. 2010. "Understanding mechanisms underlying human gene expression variation with RNA sequencing." *Nature* 464 (7289): 768–72.