# Reproduce SVA problem

Christopher Lo

9/5/2022

## Setup

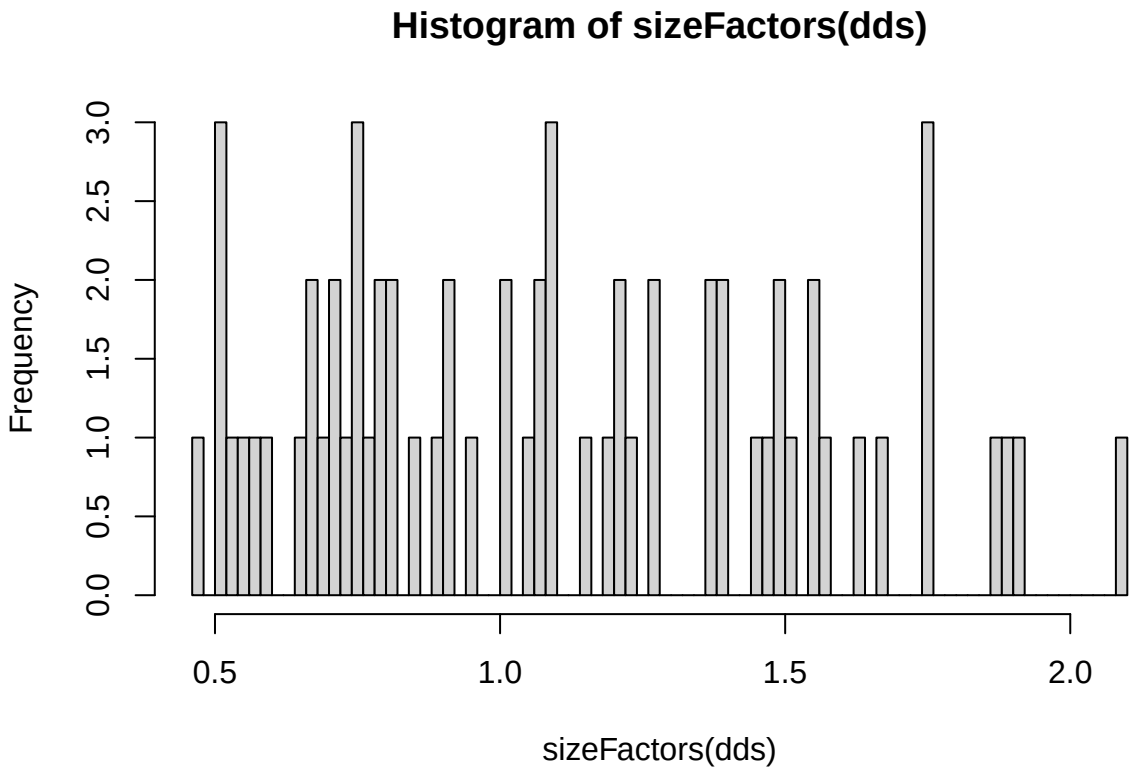
*What happens to the number of Surrogate Variables as we increase the number of studies in recount3?*

To illustrate this, we examined 4 studies that involved case/control prostate tumor and normal samples, with a minimal of 25 samples per study. For each study, we examined the number of SVs (using method = "be"), and we also aggregrated them one by one and examined the number of SVs iteratively. When using method = "leek", we always get 0 estimated SVs, which is strange.

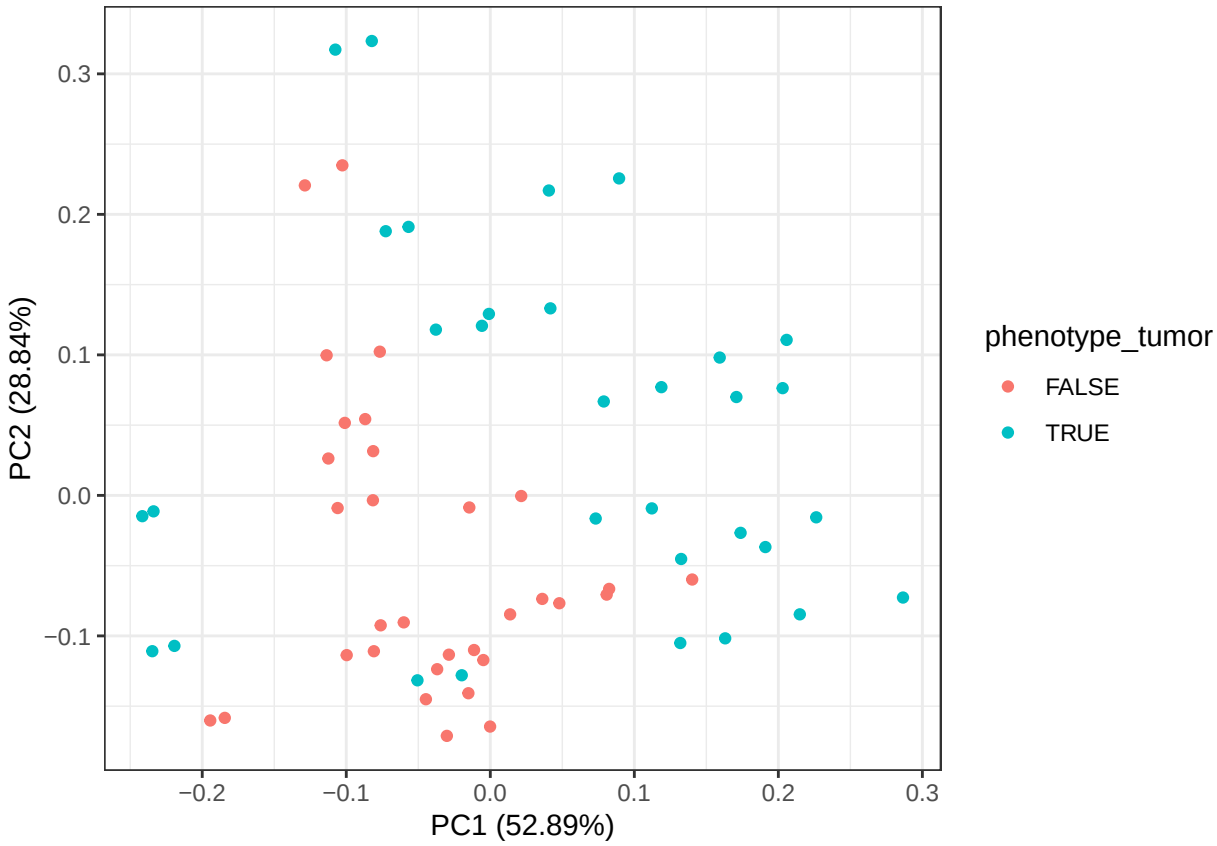
1. SRP118614: "Overall design: Matched high-grade (GS=7(4+3)) prostate tumor and adjacent normal specimens from 16 patients (8 AAM and 8 EAM) were subjected to two replicate runs of RNA-sequencing."
2. SRP212704: "Overall design: Strand specific total RNA seq was performed using frozen patient matched prostate cancer tissue in biological duplicates. Purpose: The goal of present study is to compare transcript level changes between normal and tumor of same individuals"
3. SRP002628: "Overall design: We sequenced the transcriptome (polyA+) of 20 prostate cancer tumors and 10 matched normal tissues using Illumina GAII platform. Then we used bioinformatic approaches to identify prostate cancer specific aberrations which include gene fusion, alternative splicing, somatic mutation."
4. SRP027258: "We utilized RNA sequencing to test the hypothesis that SFN modifies the expression of genes that are critical in prostate cancer progression. Normal prostate epithelial cells, and androgen-dependent and androgen-independent prostate cancer cells were treated with 15 µM SFN and the transcriptome was determined at 6 and 24 hour time points."

```
#1.
#Overall design: Matched high-grade (GS=7(4+3)) prostate tumor and adjacent
#normal specimens from 16 patients (8 AAM and 8 EAM) were subjected to two
#replicate runs of RNA-sequencing.

proj_info <- subset(
  human_projects,
  project == "SRP118614"
)

rse_gene_SRP118614 <- create_rse(proj_info)
phenotype = data.frame(colData(rse_gene_SRP118614))
genotype = data.frame(rowData(rse_gene_SRP118614))

phenotype$phenotype_tumor = grepl("prostate tumor", phenotype$sra.sample_attributes)
```

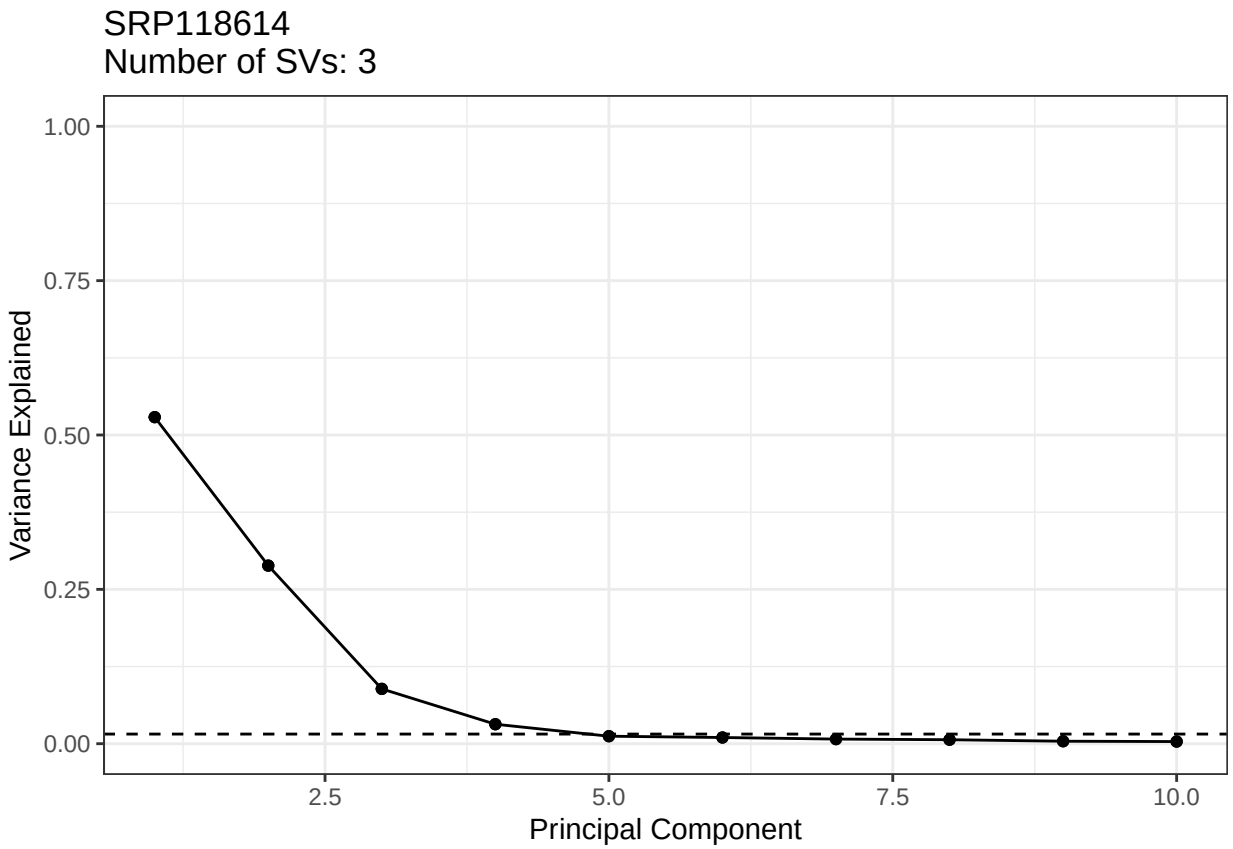
```
mod_SRP118614 = model.matrix(~phenotype_tumor, data = phenotype)

geneCounts_SRP118614 = assays(rse_gene_SRP118614)$raw_counts
dds <- DESeqDataSetFromMatrix(countData = geneCounts_SRP118614, colData = phenotype, design = ~ phenoty
dds <- estimateSizeFactors(dds)
hist(sizeFactors(dds), 100)
```




```
geneCounts_SRP118614 <- counts(dds, normalized=TRUE)
#geneCounts_SRP118614 = log(geneCounts_SRP118614 / colSums(geneCounts_SRP118614) + 5) #normalize by bas

n_sv[1] = num.sv(geneCounts_SRP118614, mod_SRP118614, method = "be", vfilter = 10000)
n_sv_agg[1] = NA

pca = prcomp(t(geneCounts_SRP118614))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]
autoplot(pca, data = phenotype, colour = 'phenotype_tumor')
```

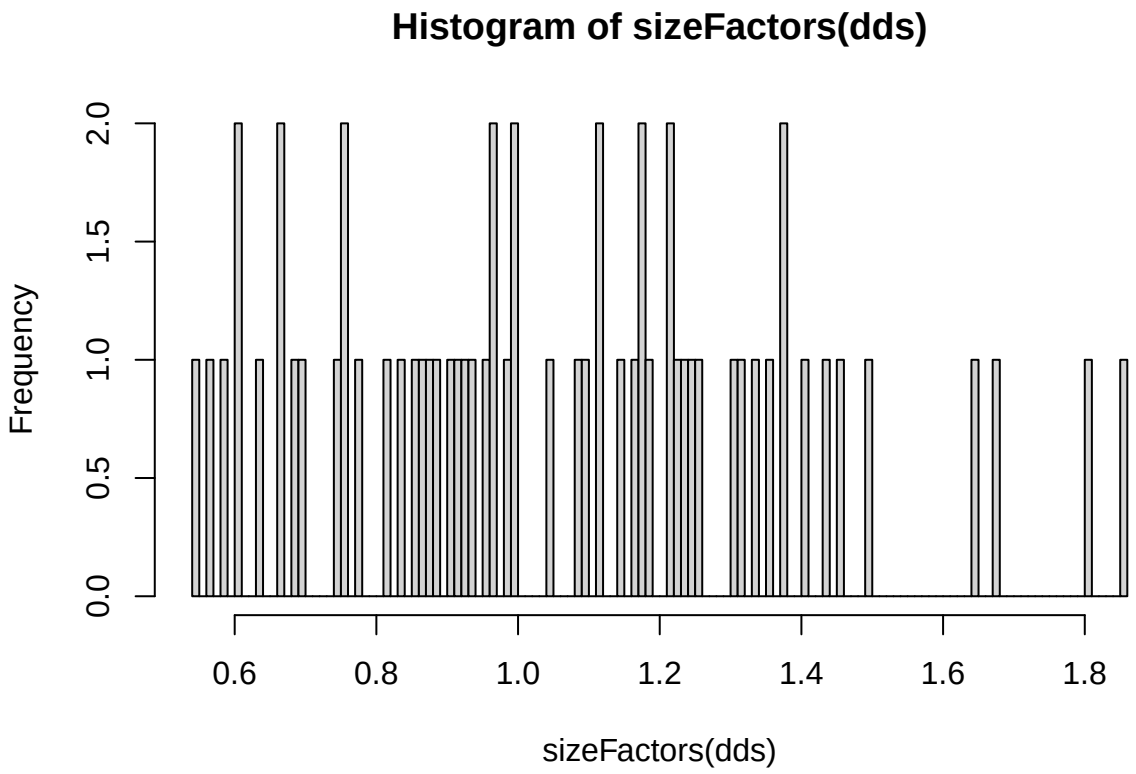



```
qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_SRP118614), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP118614 \nNumber of SVs:
```

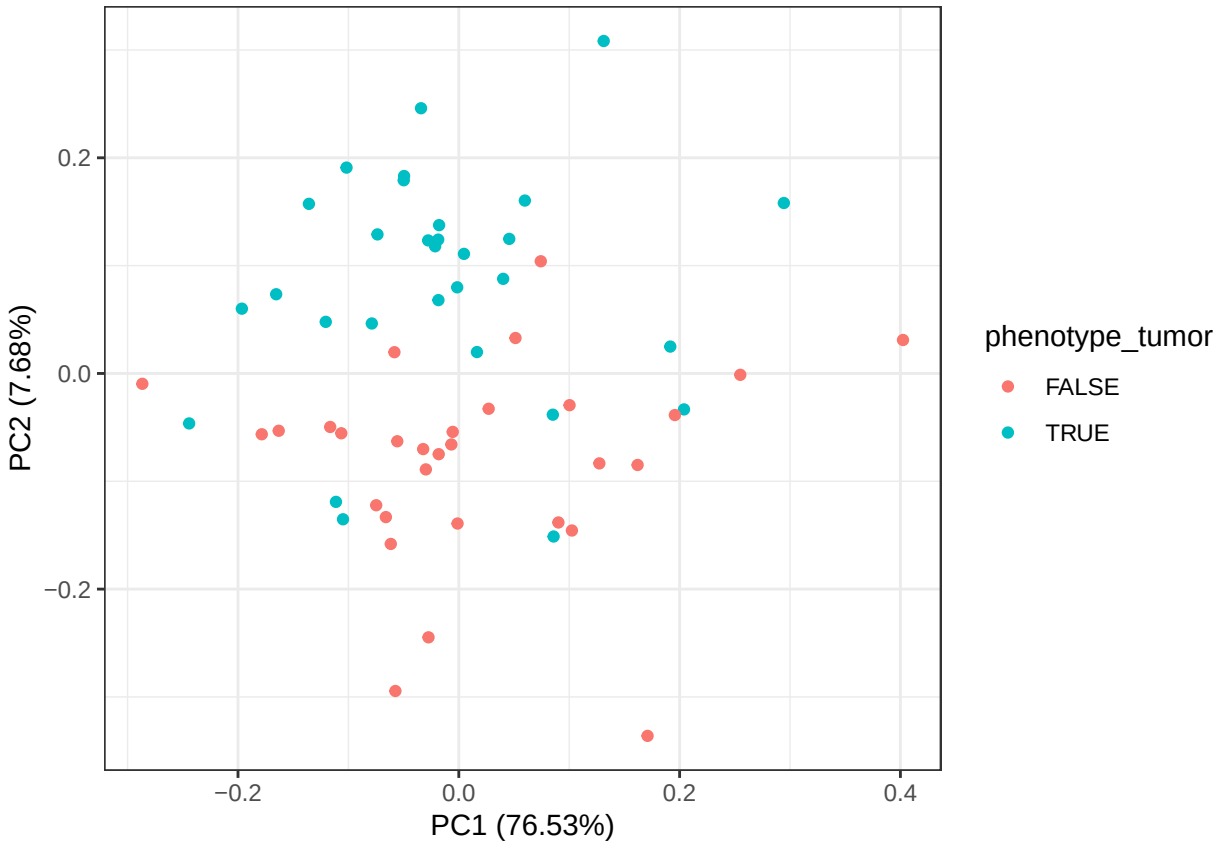



Our scree plot threshold lines up with our num.sv estimates quite well here.

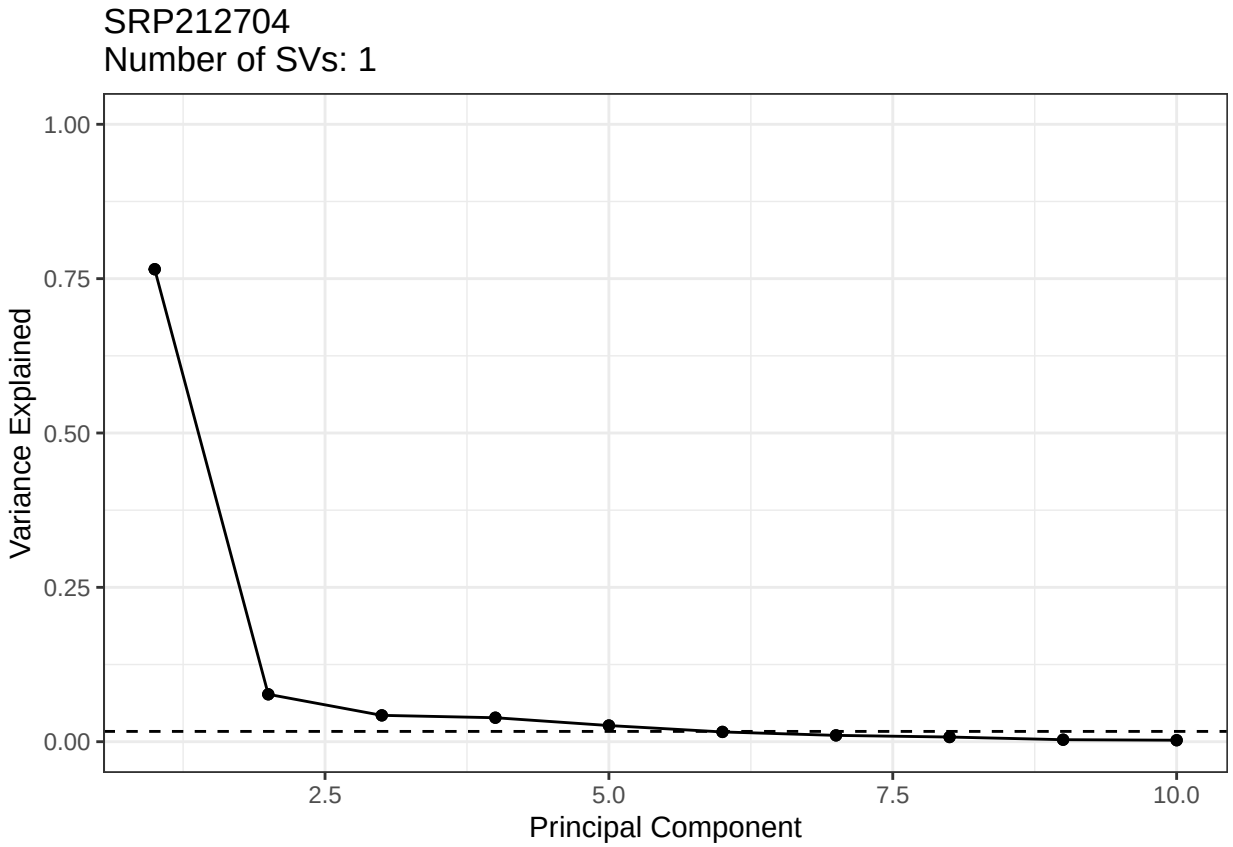
```
#2.
#Overall design: Strand specific total RNA seq was performed using frozen
#patient matched prostate cancer tissue in biological duplicates.
#Purpose: The goal of present study is to compare transcript level changes
#between normal and tumor of same individuals

proj_info <- subset(
  human_projects,
  project == "SRP212704"
)

rse_gene_SRP212704 <- create_rse(proj_info)
phenotype = data.frame(colData(rse_gene_SRP212704))
genotype = data.frame(rowData(rse_gene_SRP212704))

phenotype$phenotype_tumor = grepl("Tumor", phenotype$sra.sample_attributes)
mod_SRP212704 = model.matrix(~phenotype_tumor, data = phenotype)

geneCounts_SRP212704 = assays(rse_gene_SRP212704)$raw_counts
dds <- DESeqDataSetFromMatrix(countData = geneCounts_SRP212704, colData = phenotype, design = ~ phenoty
dds <- estimateSizeFactors(dds)
hist(sizeFactors(dds), 100)
```



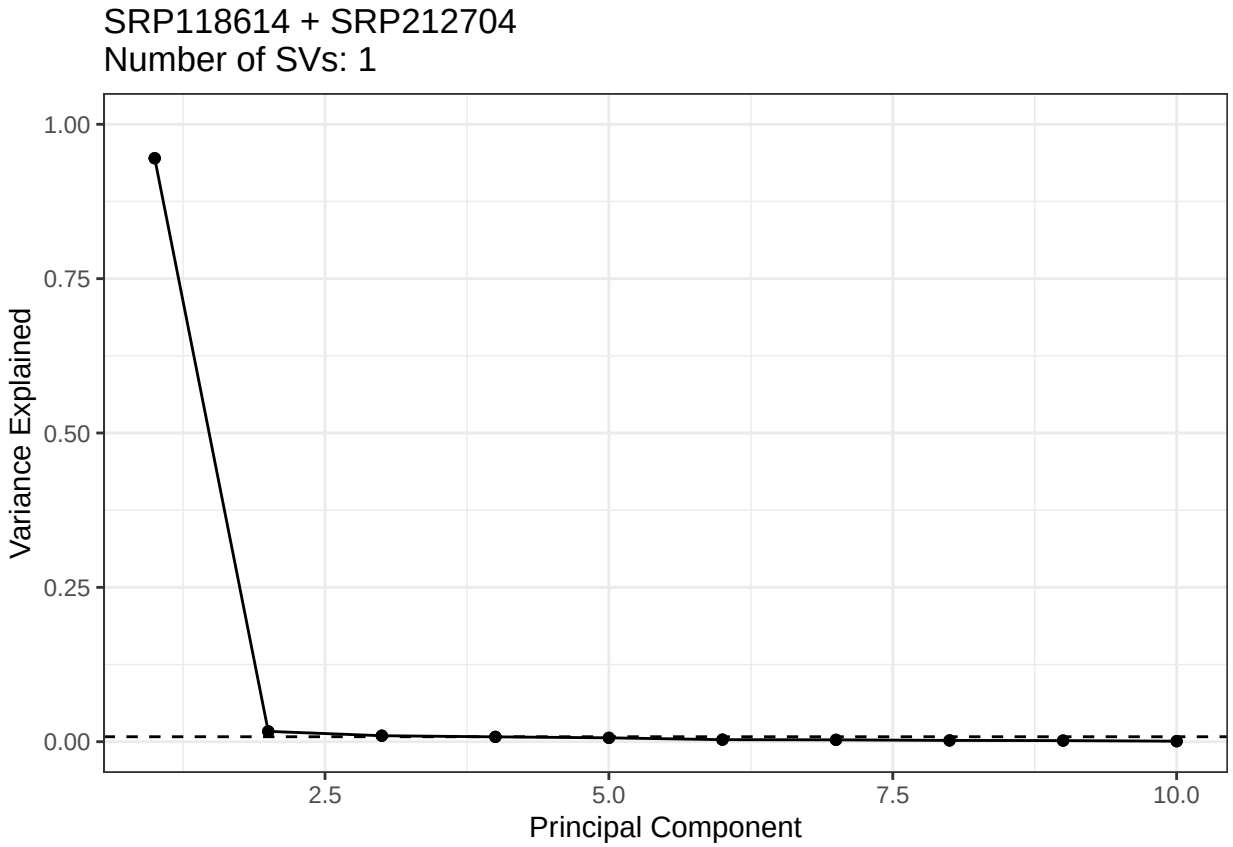

```
geneCounts_SRP212704 <- counts(dds, normalized=TRUE)

geneCounts_2 = cbind(geneCounts_SRP118614, geneCounts_SRP212704)
mod_2 = rbind(mod_SRP118614, mod_SRP212704)
mod_2_ext = as.data.frame(mod_2)
mod_2_ext$study = c(rep(0, ncol(geneCounts_SRP118614)),
             rep(1, ncol(geneCounts_SRP212704)))

n_sv[2] = num.sv(geneCounts_SRP212704, mod_SRP212704, method = "be", vfilter = 10000)
n_sv_agg[2] = num.sv(geneCounts_2, as.matrix(mod_2_ext), method = "be", vfilter = 10000)

pca = prcomp(t(geneCounts_SRP212704))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]
autoplot(pca, data = phenotype, colour = 'phenotype_tumor')
```



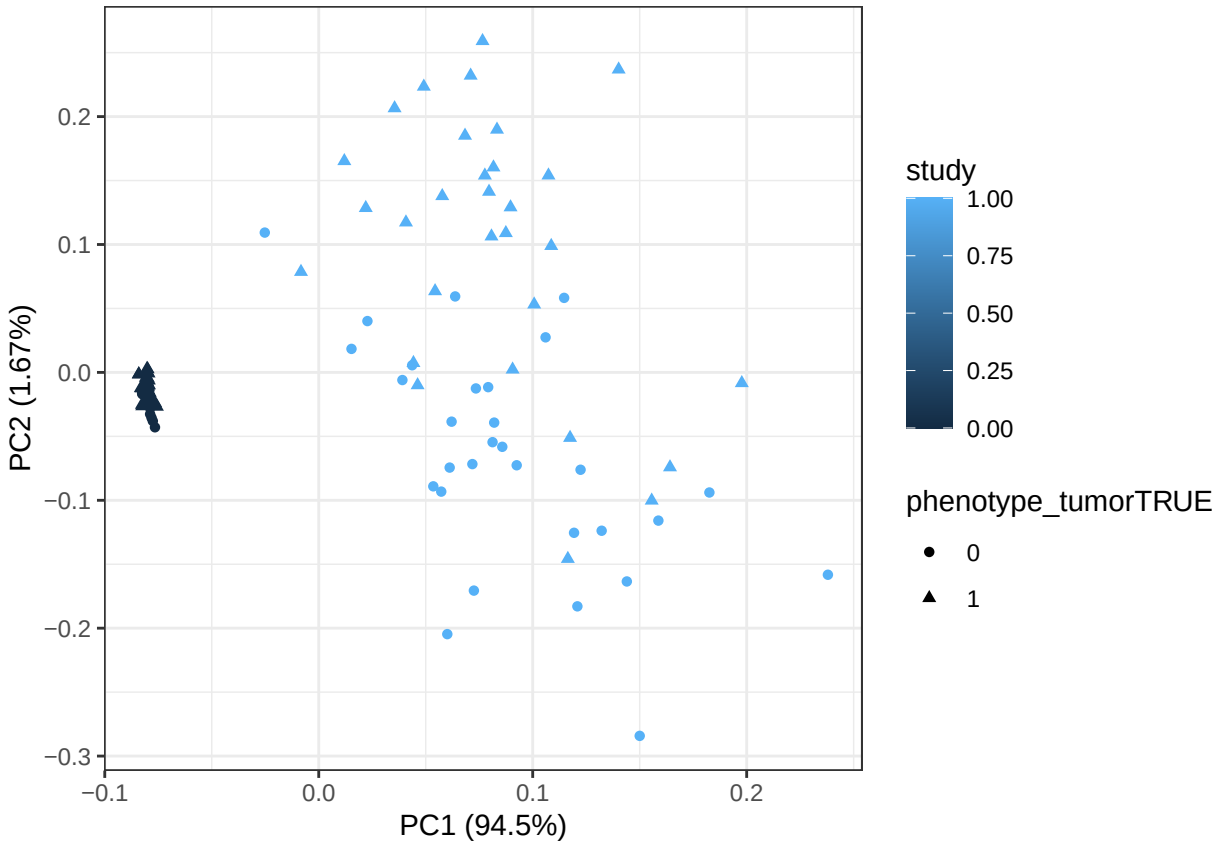

```
qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_SRP212704), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP212704 \nNumber of SVs:
```



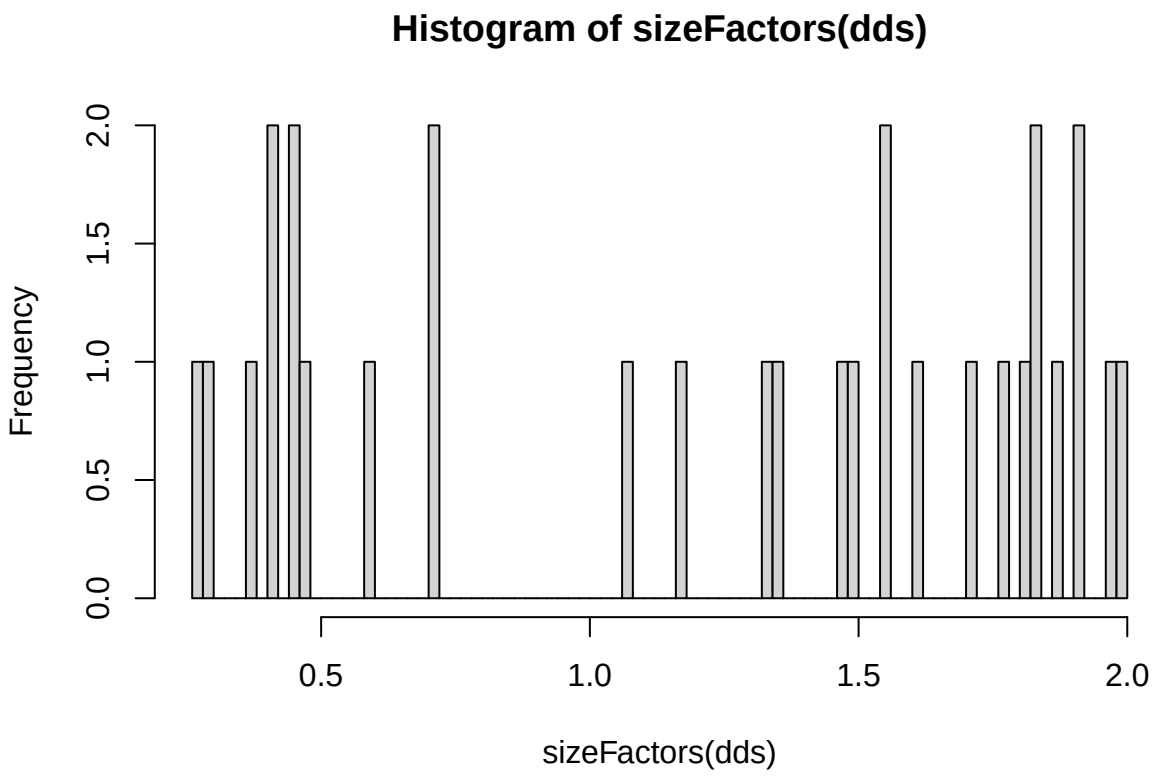

```
pca = prcomp(t(geneCounts_2))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]

qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_2), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP118614 + SRP212704 \nNu
```



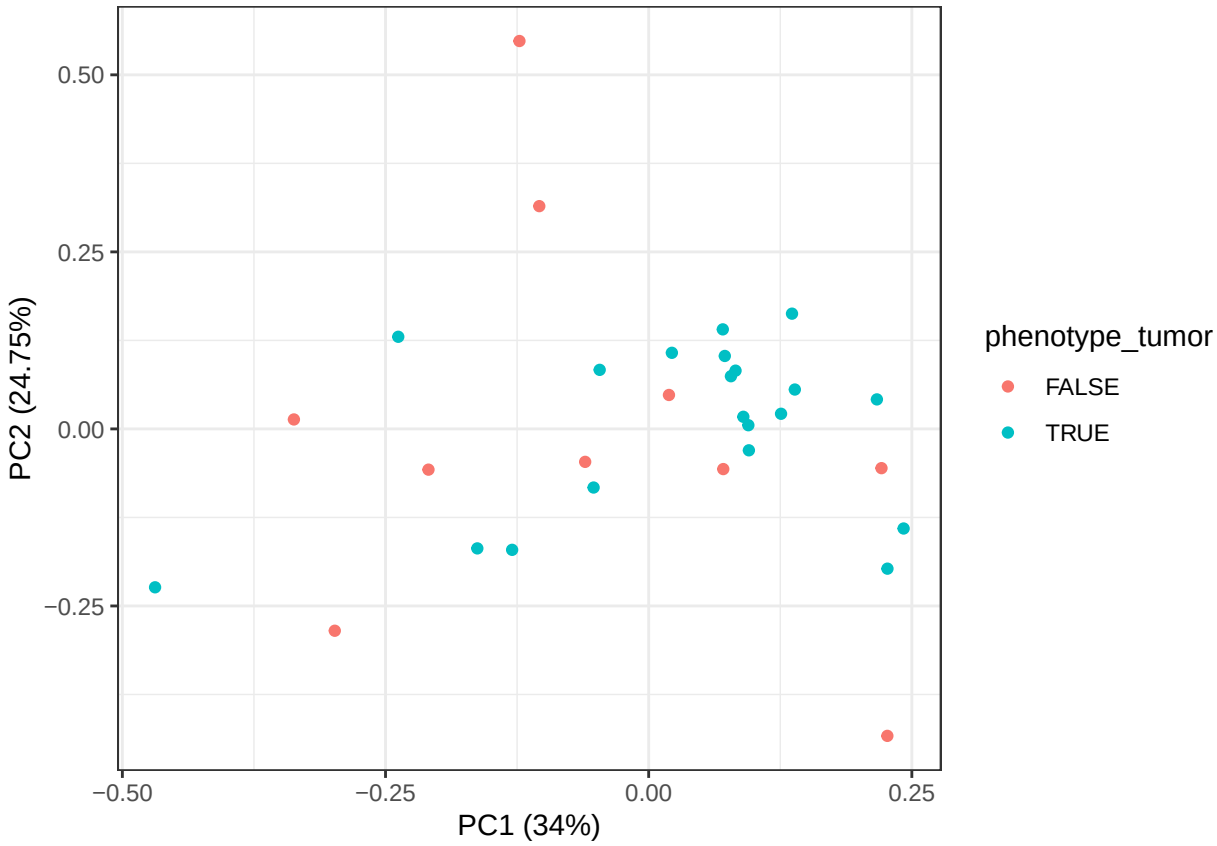

```
mod_2_ext = as.data.frame(mod_2)
mod_2_ext$phenotype_tumorTRUE = as.factor(mod_2_ext$phenotype_tumorTRUE)
mod_2_ext$study = c(rep(0, ncol(geneCounts_SRP118614)),
             rep(1, ncol(geneCounts_SRP212704)))
autoplot(pca, data = mod_2_ext, colour = 'study', shape = 'phenotype_tumorTRUE')
```



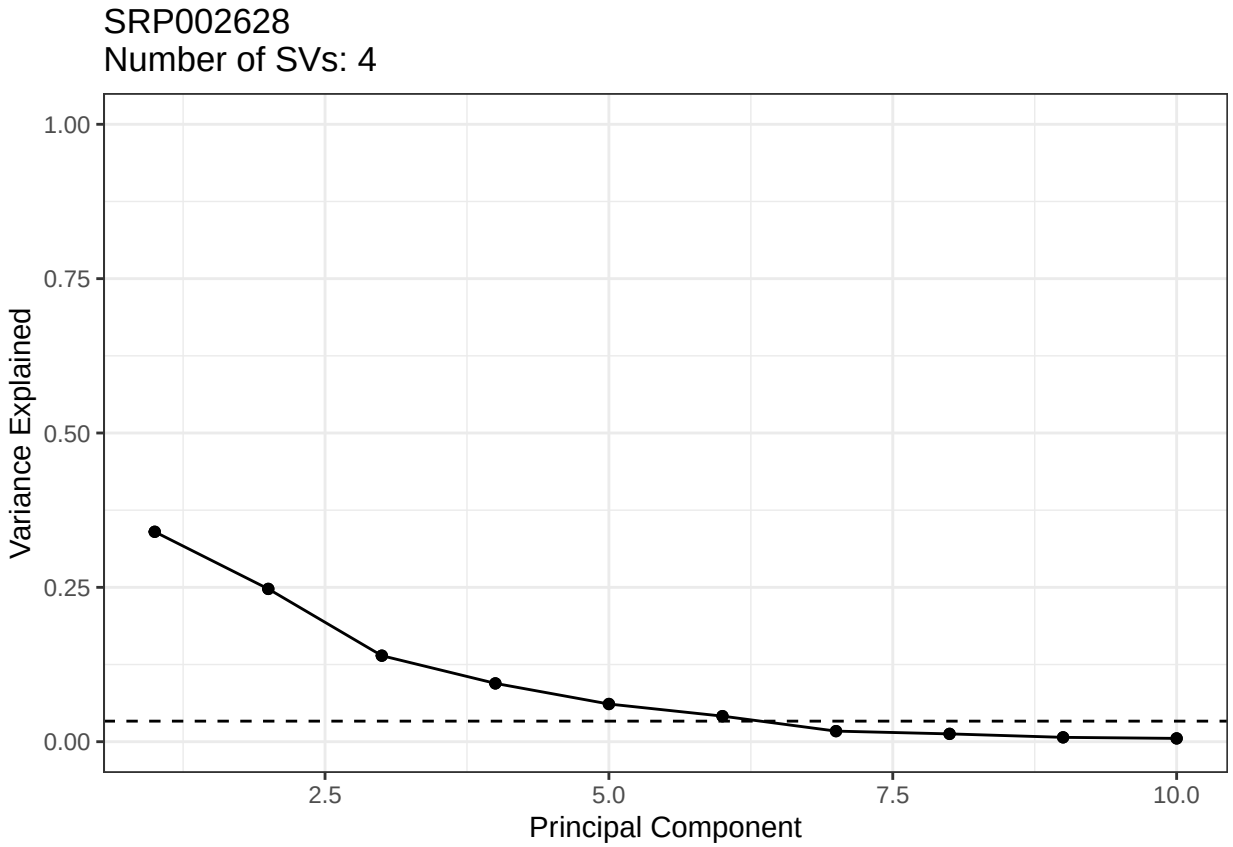

```
#3
#Overall design: We sequenced the transcriptome (polyA+) of 20 prostate cancer
#tumors and 10 matched normal tissues using Illumina GAII platform.
#Then we used bioinformatic approaches to identify prostate cancer specific
#aberrations which include gene fusion, alternative splicing, somatic mutation
proj_info <- subset(
  human_projects,
  project == "SRP002628"
)
rse_gene_SRP002628 <- create_rse(proj_info)
phenotype = data.frame(colData(rse_gene_SRP002628))
genotype = data.frame(rowData(rse_gene_SRP002628))

phenotype$phenotype_tumor = grepl("Prostate cancer tissue", phenotype$sra.sample_attributes)
mod_SRP002628 = model.matrix(~phenotype_tumor, data = phenotype)

geneCounts_SRP002628 = assays(rse_gene_SRP002628)$raw_counts
dds <- DESeqDataSetFromMatrix(countData = geneCounts_SRP002628, colData = phenotype, design = ~ phenoty
dds <- estimateSizeFactors(dds)
hist(sizeFactors(dds), 100)
```



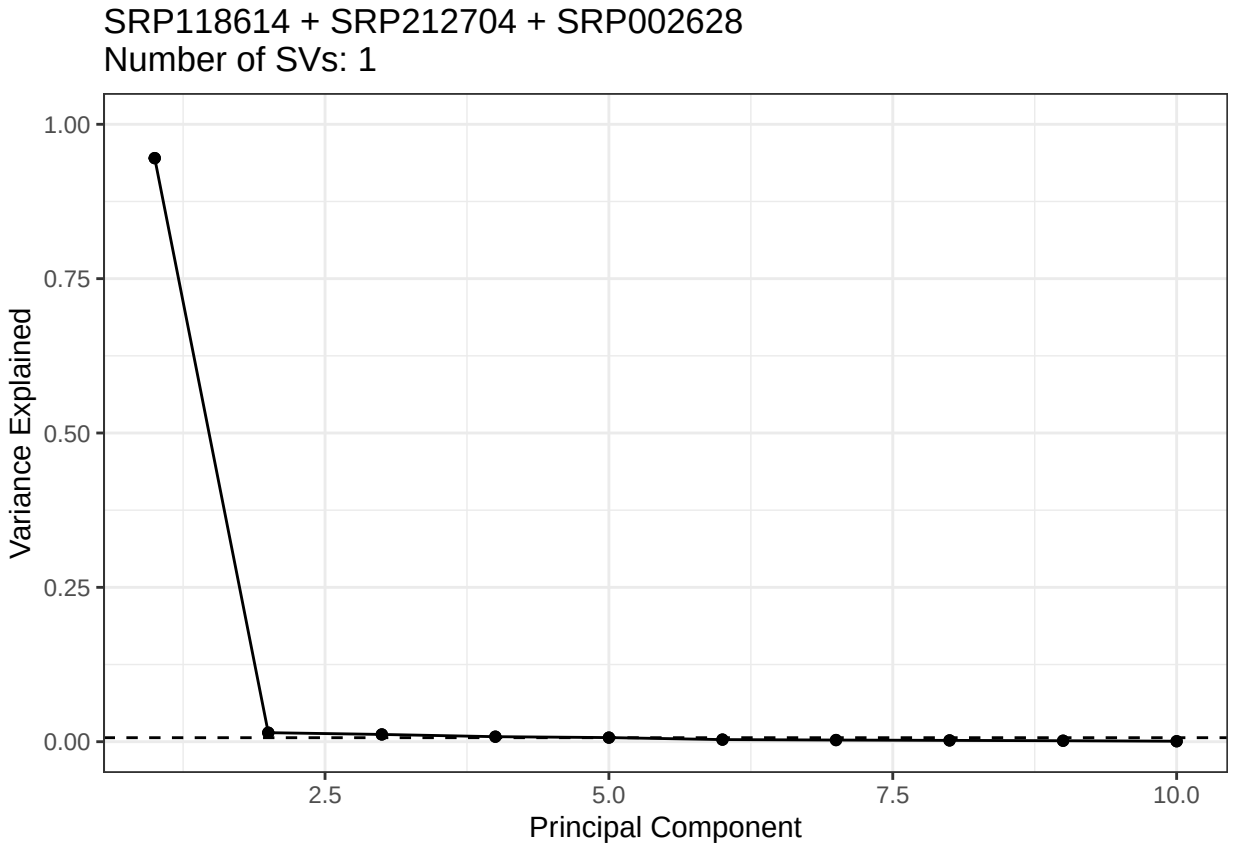

```
geneCounts_SRP002628 <- counts(dds, normalized=TRUE)


geneCounts_3 = cbind(geneCounts_SRP118614, geneCounts_SRP212704, geneCounts_SRP002628)
mod_3 = rbind(mod_SRP118614, mod_SRP212704, mod_SRP002628)
mod_3_ext = as.data.frame(mod_3)
study = c(rep(0, ncol(geneCounts_SRP118614)),
          rep(1, ncol(geneCounts_SRP212704)),
          rep(2, ncol(geneCounts_SRP002628)))
mod_3_ext$study2 = ifelse(study == 1, 1, 0)
mod_3_ext$study3 = ifelse(study == 2, 1, 0)

n_sv[3] = num.sv(geneCounts_SRP002628, mod_SRP002628, method = "be", vfilter = 10000)
n_sv_agg[3] = num.sv(geneCounts_3, as.matrix(mod_3_ext), method = "be", vfilter = 10000)


pca = prcomp(t(geneCounts_SRP002628))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]
autoplot(pca, data = phenotype, colour = 'phenotype_tumor')
```



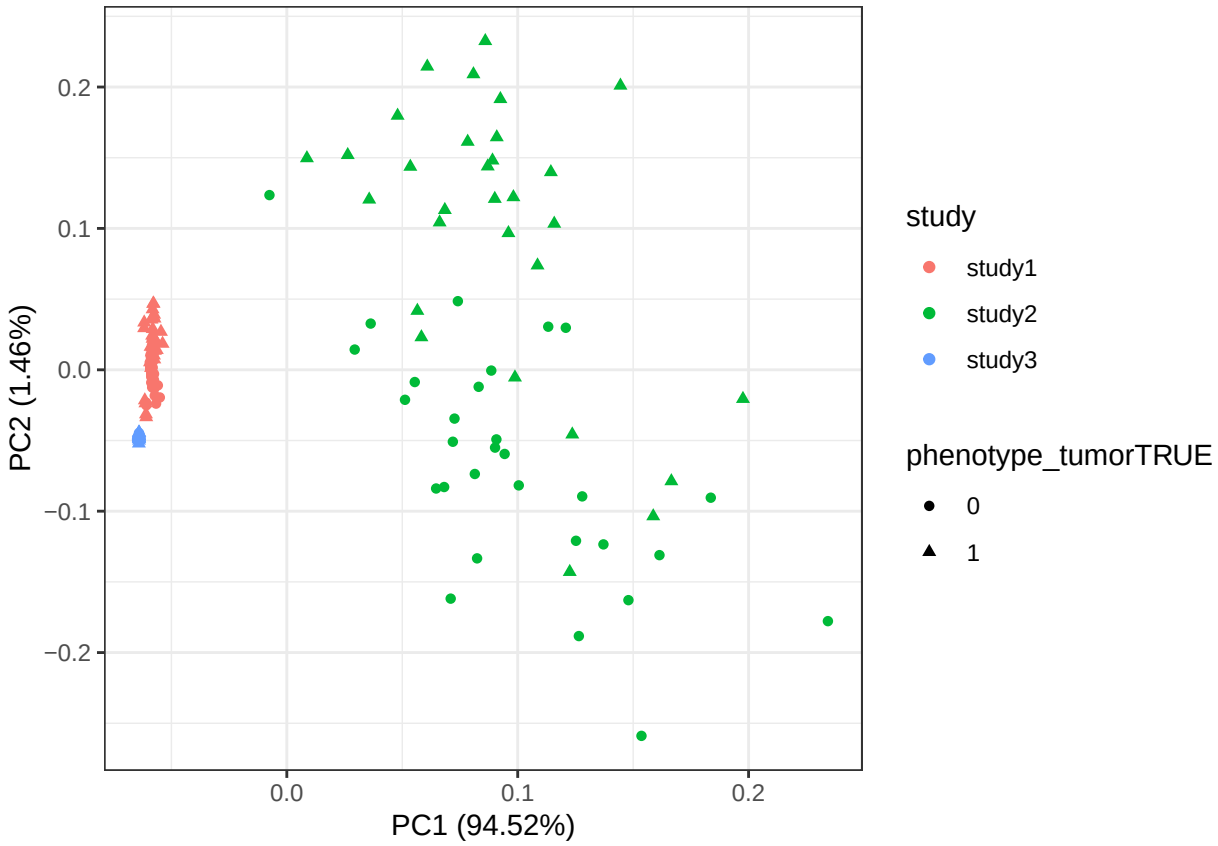

```
qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_SRP002628), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP002628 \nNumber of SVs:
```



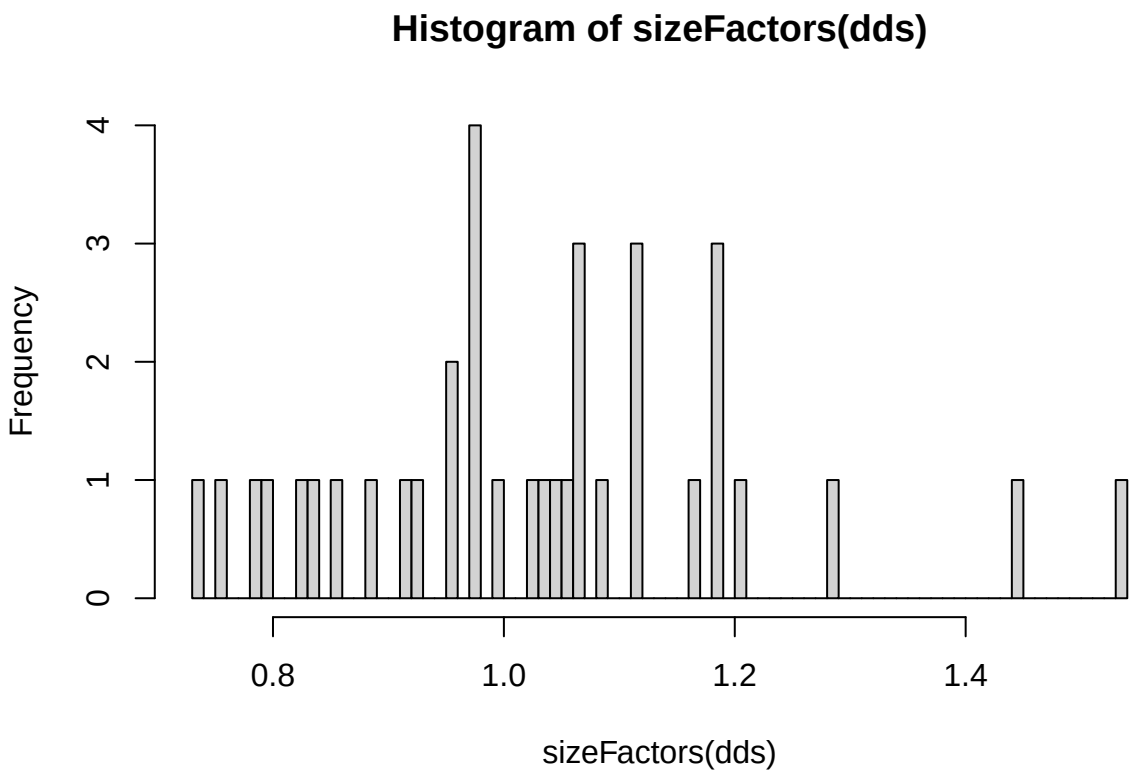

```
pca = prcomp(t(geneCounts_3))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]

qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_3), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP118614 + SRP212704 + SR
```



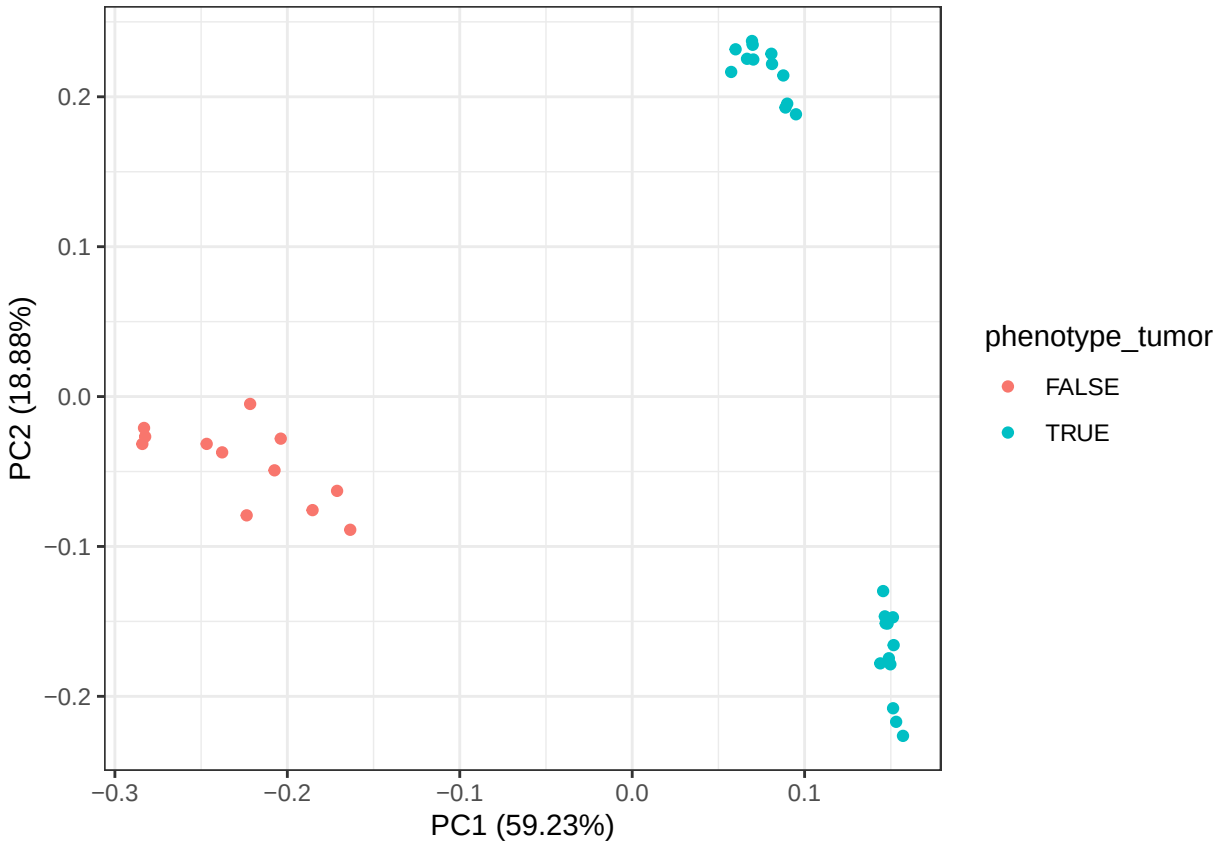

```
mod_3_ext = as.data.frame(mod_3)
mod_3_ext$phenotype_tumorTRUE = as.factor(mod_3_ext$phenotype_tumorTRUE)
mod_3_ext$study = c(rep("study1", ncol(geneCounts_SRP118614)),
             rep("study2", ncol(geneCounts_SRP212704)),
             rep("study3", ncol(geneCounts_SRP002628)))
autoplot(pca, data = mod_3_ext, colour = 'study', shape = 'phenotype_tumorTRUE')
```



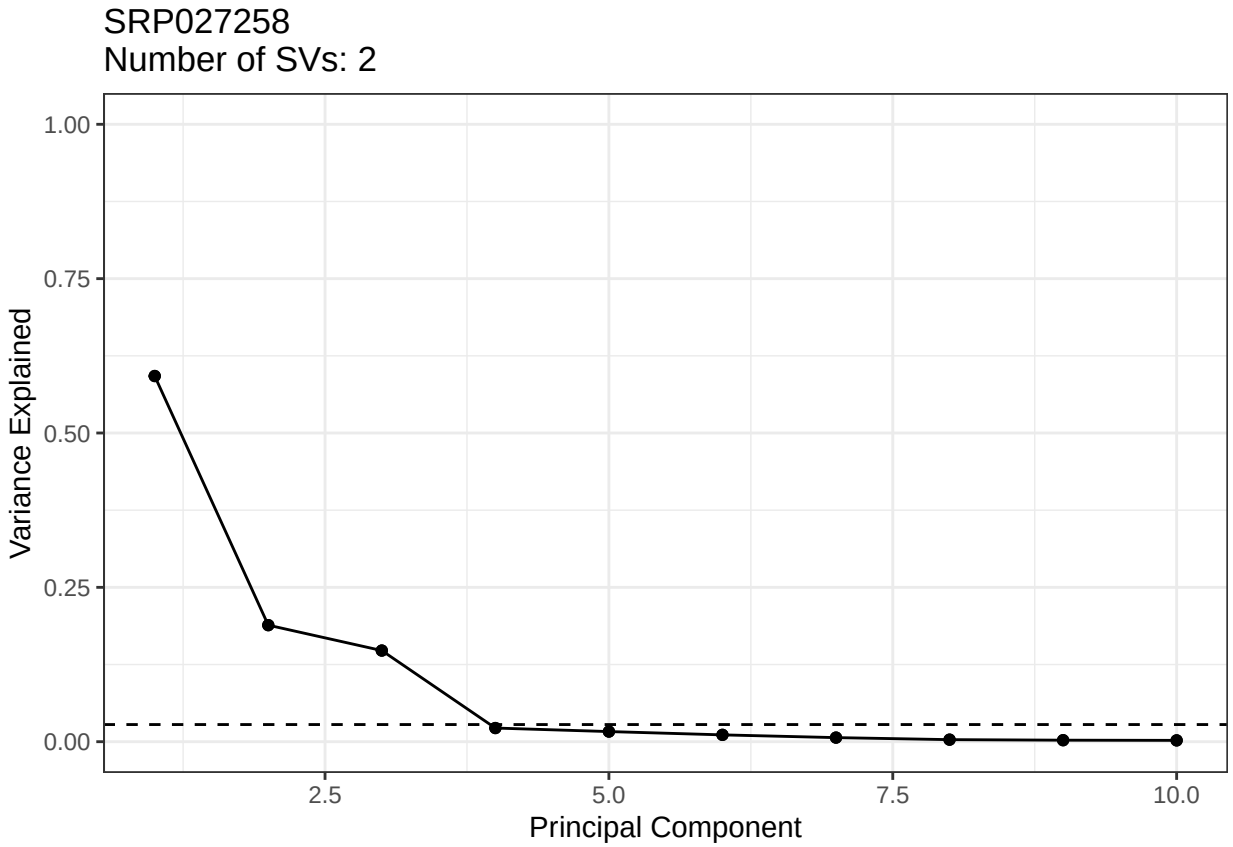

```
#4
#We utilized RNA sequencing to test the hypothesis that SFN modifies the
#expression of genes that are critical in prostate cancer progression.
#Normal prostate epithelial cells, and androgen-dependent and androgen-independent
#prostate cancer cells were treated with 15 μM SFN and the transcriptome was
#determined at 6 and 24 hour time points.
proj_info <- subset(
  human_projects,
  project == "SRP027258"
)
rse_gene_SRP027258 <- create_rse(proj_info)
phenotype = data.frame(colData(rse_gene_SRP027258))
genotype = data.frame(rowData(rse_gene_SRP027258))

phenotype$phenotype_tumor = grepl("prostate cancer", phenotype$sra.sample_attributes)
mod_SRP027258 = model.matrix(~phenotype_tumor, data = phenotype)


geneCounts_SRP027258 = assays(rse_gene_SRP027258)$raw_counts
dds <- DESeqDataSetFromMatrix(countData = geneCounts_SRP027258, colData = phenotype, design = ~ phenoty
dds <- estimateSizeFactors(dds)
hist(sizeFactors(dds), 100)
```



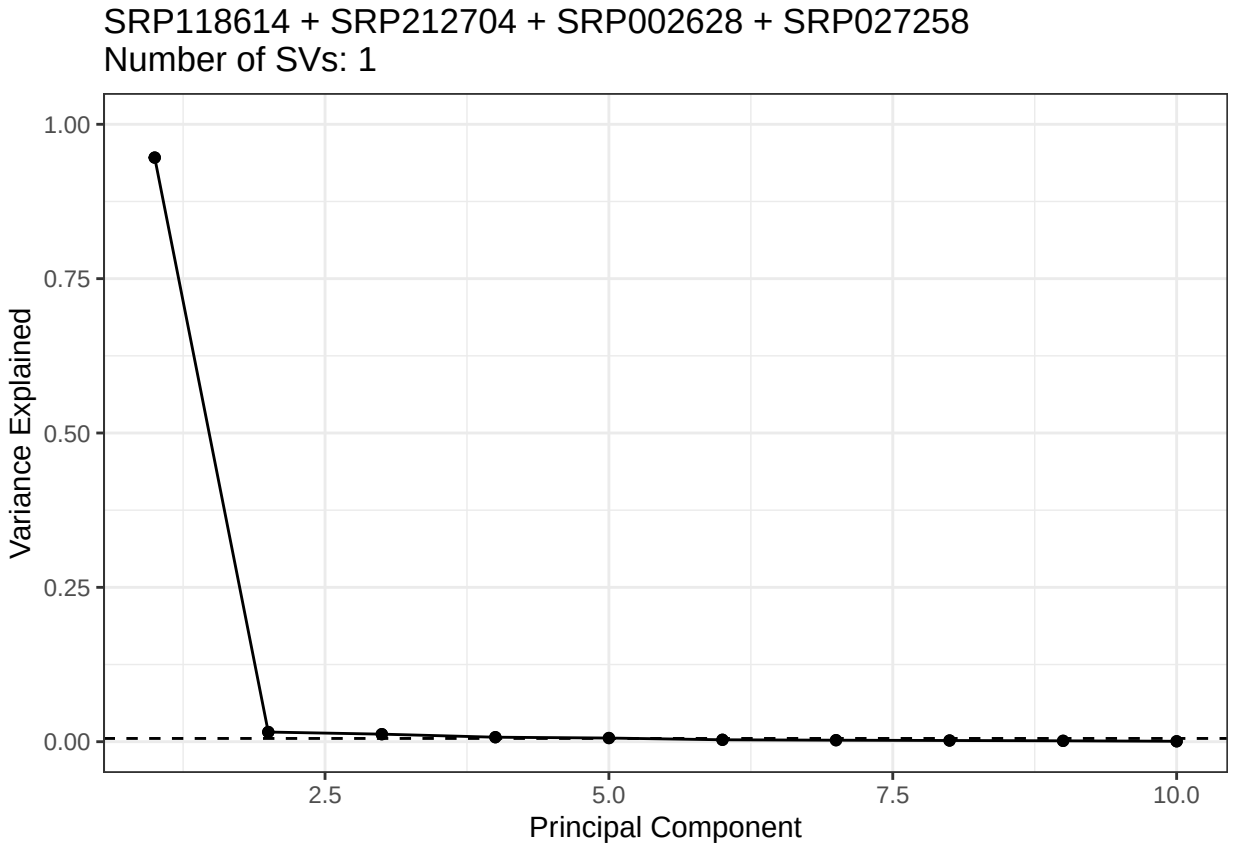

```
geneCounts_SRP027258 <- counts(dds, normalized=TRUE)


geneCounts_4 = cbind(geneCounts_SRP118614, geneCounts_SRP212704, geneCounts_SRP002628, geneCounts_SRP02
mod_4 = rbind(mod_SRP118614, mod_SRP212704, mod_SRP002628, mod_SRP027258)
mod_4_ext = as.data.frame(mod_4)
study = c(rep(0, ncol(geneCounts_SRP118614)),
          rep(1, ncol(geneCounts_SRP212704)),
          rep(2, ncol(geneCounts_SRP002628)),
          rep(3, ncol(geneCounts_SRP027258)))
mod_4_ext$study2 = ifelse(study == 1, 1, 0)
mod_4_ext$study3 = ifelse(study == 2, 1, 0)

n_sv[4] = num.sv(geneCounts_SRP027258, mod_SRP027258, method = "be", vfilter = 10000)
n_sv_agg[4] = num.sv(geneCounts_4, as.matrix(mod_4_ext), method = "be", vfilter = 10000)


pca = prcomp(t(geneCounts_SRP027258))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]
autoplot(pca, data = phenotype, colour = 'phenotype_tumor')
```



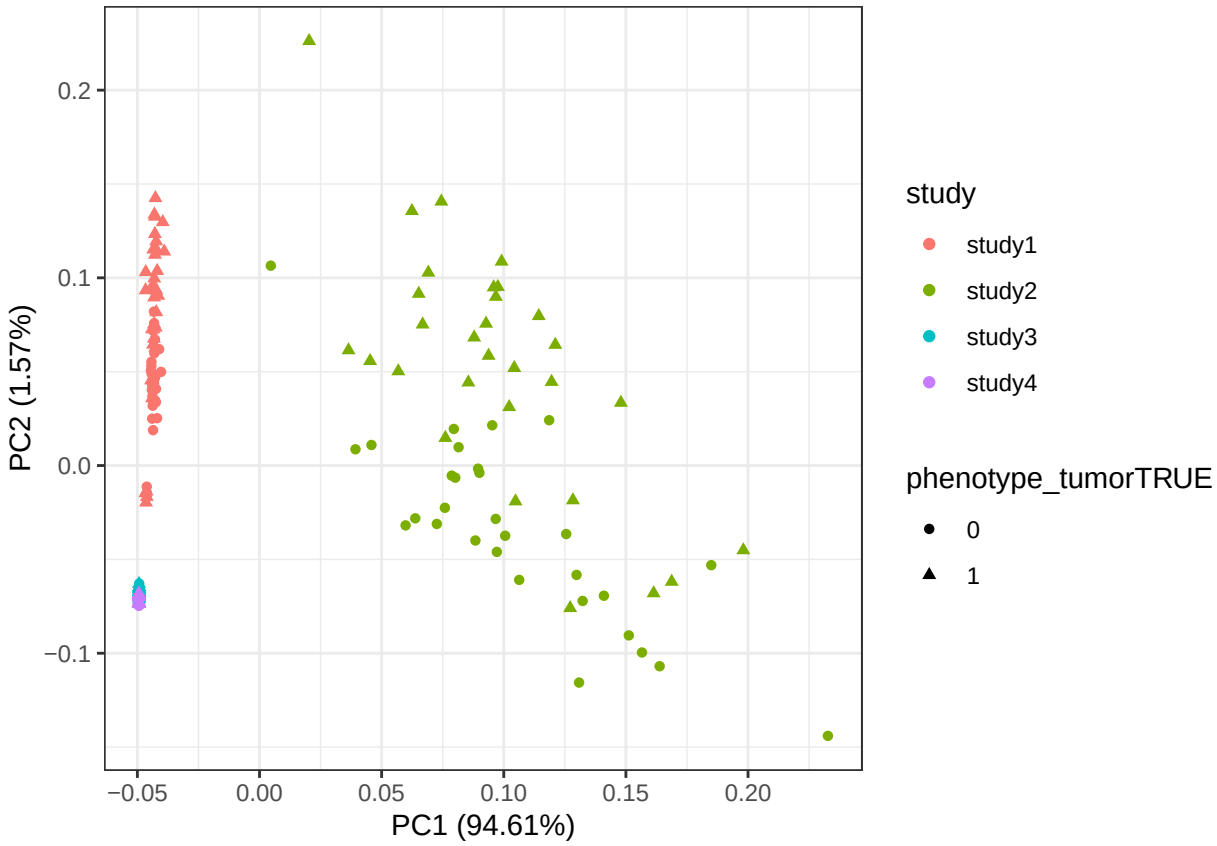

```
qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_SRP027258), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP027258 \nNumber of SVs:
```




```
pca = prcomp(t(geneCounts_4))
variance = pca$sdev^2 / sum(pca$sdev^2)
variance = variance[1:10]

qplot(c(1:length(variance)), variance) + geom_line() + geom_point() +
  geom_hline(yintercept=1/ncol(geneCounts_4), linetype = "dashed") +
  xlab("Principal Component") + ylab("Variance Explained") + ggtitle(paste0("SRP118614 + SRP212704 + SRI
```

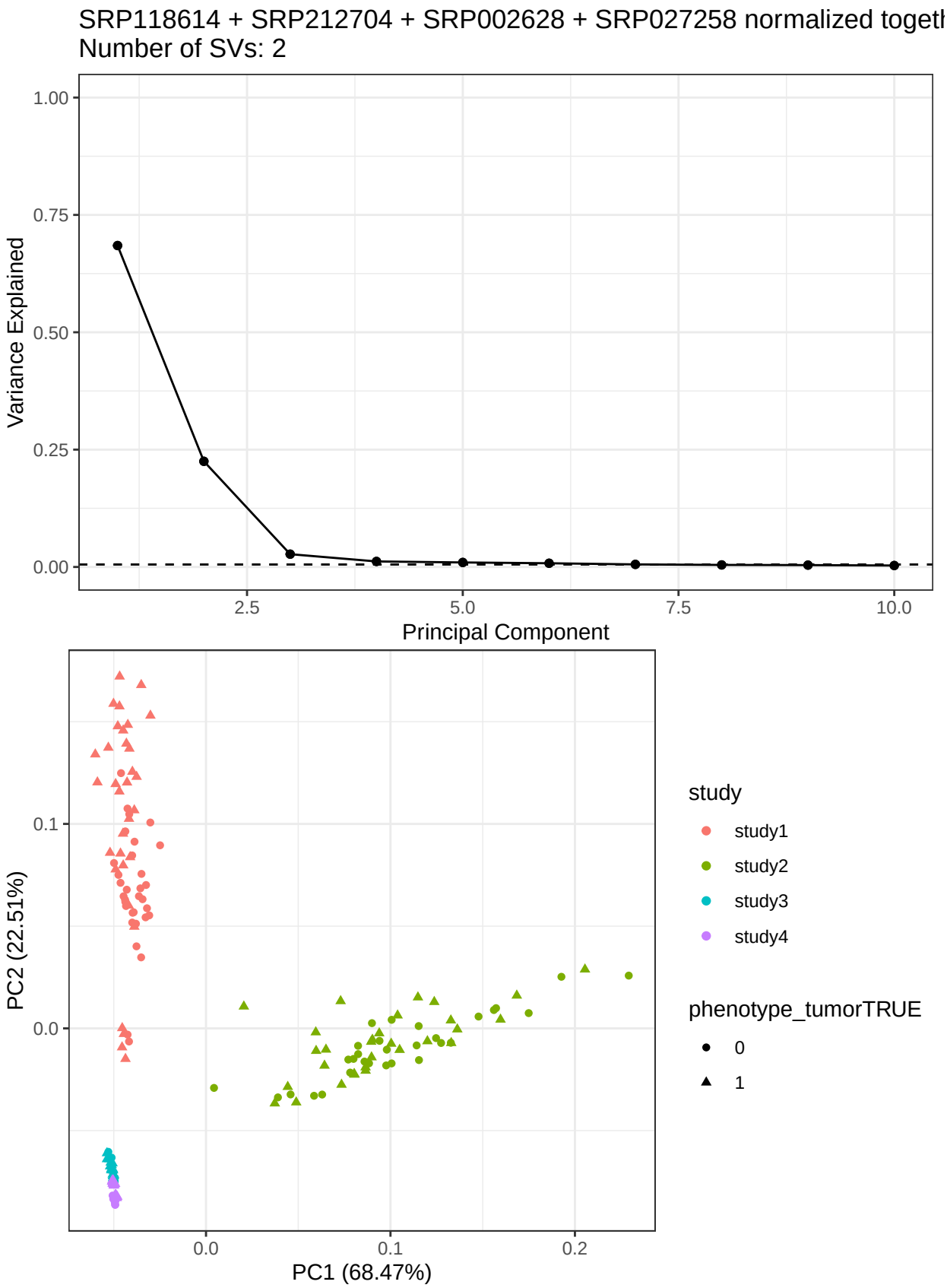



```
mod_4_ext = as.data.frame(mod_4)
mod_4_ext$phenotype_tumorTRUE = as.factor(mod_4_ext$phenotype_tumorTRUE)
mod_4_ext$study = c(rep("study1", ncol(geneCounts_SRP118614)),
             rep("study2", ncol(geneCounts_SRP212704)),
             rep("study3", ncol(geneCounts_SRP002628)),
             rep("study4", ncol(geneCounts_SRP027258)))
autoplot(pca, data = mod_4_ext, colour = 'study', shape = 'phenotype_tumorTRUE')
```




What if we combine all the studies together before normalization?

```
## converting counts to integer mode
```


SRP118614 + SRP212704 + SRP002628 + SRP027258 normalized togetl
Number of SVs: 2
Variance Explained
1.00
0.75
0.50
0.25
0.00
2.5
5.0
7.5
10.0
Principal Component
PC2 (22.51%)
0.1
0.0
0.0
0.1
0.2
PC1 (68.47%)
study
study1
study2
study3
study4
phenotype_tumorTRUE
0
1

## Results

The number of SVs for each study: 3, 1, 4, 2.

The number of SVs for each aggregration (study 1+2, 1+2+3, 1+2+3+4): NA, 1, 1, 1.

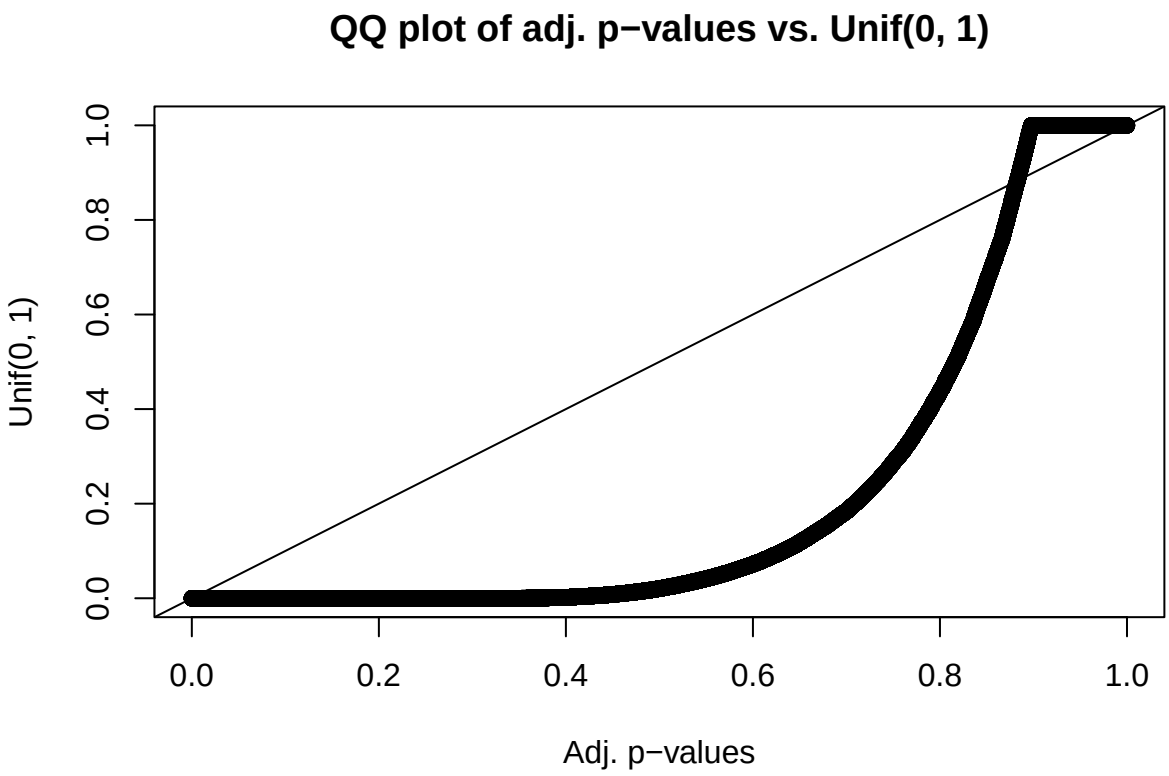
## Follow-up

We then consider the following: Is there a difference between study 1 and study 2 just using control samples with SVs? When we run DE, we would expect theadj. p-values look uniform.

```
#We then consider the following:
#Is there a difference between study 1 and study 2 just using control samples with SVs?
#Run DE: Do adj. p-values look uniform?
mod_2controls = c(rep(0, ncol(geneCounts_SRP118614)),
                  rep(1, ncol(geneCounts_SRP212704)))
controls = which(mod_2[, 2] == 0)
geneCounts_2controls = geneCounts_2[, controls]
mod_2controls = mod_2controls[controls]
mod_2controls = model.matrix(~ mod_2controls)

n.sv = num.sv(geneCounts_2controls, mod_2controls, method = "be", vfilter = 10000)
svobj = sva(geneCounts_2controls, mod_2controls, mod_2controls[, 1], n.sv = n.sv, vfilter = 10000)
```

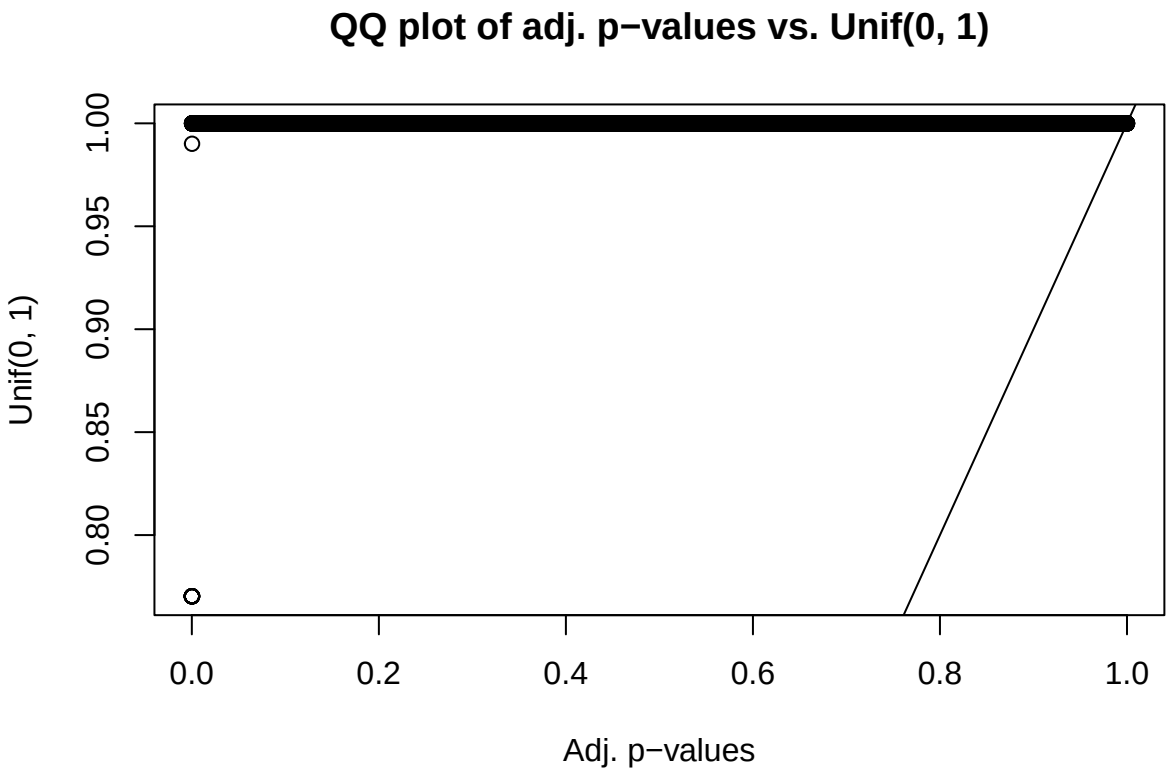
```
## Number of significant surrogate variables is:  1
## Iteration (out of 5 ):1  2  3  4  5
```

```
mod_2controls_sv = cbind(mod_2controls, svobj$sv)
#run limma for DE
fit = lmFit(geneCounts_2controls, mod_2controls_sv)
eb = eBayes(fit)
result = topTable(eb, adjust = "BH", number = nrow(geneCounts_2controls))

plot(1:nrow(result)/(nrow(result)+1), sort(result$adj.P.Val), xlab="Adj. p-values", ylab="Unif(0, 1)")
abline(a = 0, b = 1)
title(main="QQ plot of adj. p-values vs. Unif(0, 1)")
```




That looks weird. A even more simple case is to take control samples from study 1, split it into two random groups (sample sizes of 12 and 12), run SVA, and perform DE.

```
geneCounts_SRP118614_controls = geneCounts_SRP118614[, phenotype$phenotype_tumor == F]
case_control = rep(0, ncol(geneCounts_SRP118614_controls))
case_control[sample(1:ncol(geneCounts_SRP118614_controls), size = ncol(geneCounts_SRP118614_controls)/2)
mod = model.matrix(~ case_control)


n.sv = num.sv(geneCounts_SRP118614_controls, mod, method = "be", vfilter = 10000)
svobj = sva(geneCounts_SRP118614_controls, mod, mod[, 1], n.sv = n.sv, vfilter = 10000)
mod_sv = cbind(mod, svobj$sv)

#run limma for DE
fit = lmFit(geneCounts_SRP118614_controls, mod_sv)
eb = eBayes(fit)
result = topTable(eb, adjust = "BH", number = nrow(geneCounts_SRP118614_controls))

plot(1:nrow(result)/(nrow(result)+1), sort(result$adj.P.Val), xlab="Adj. p-values", ylab="Unif(0, 1)")
abline(a = 0, b = 1)
title(main="QQ plot of adj. p-values vs. Unif(0, 1)")
```




Looks. . . better?