Usoskin

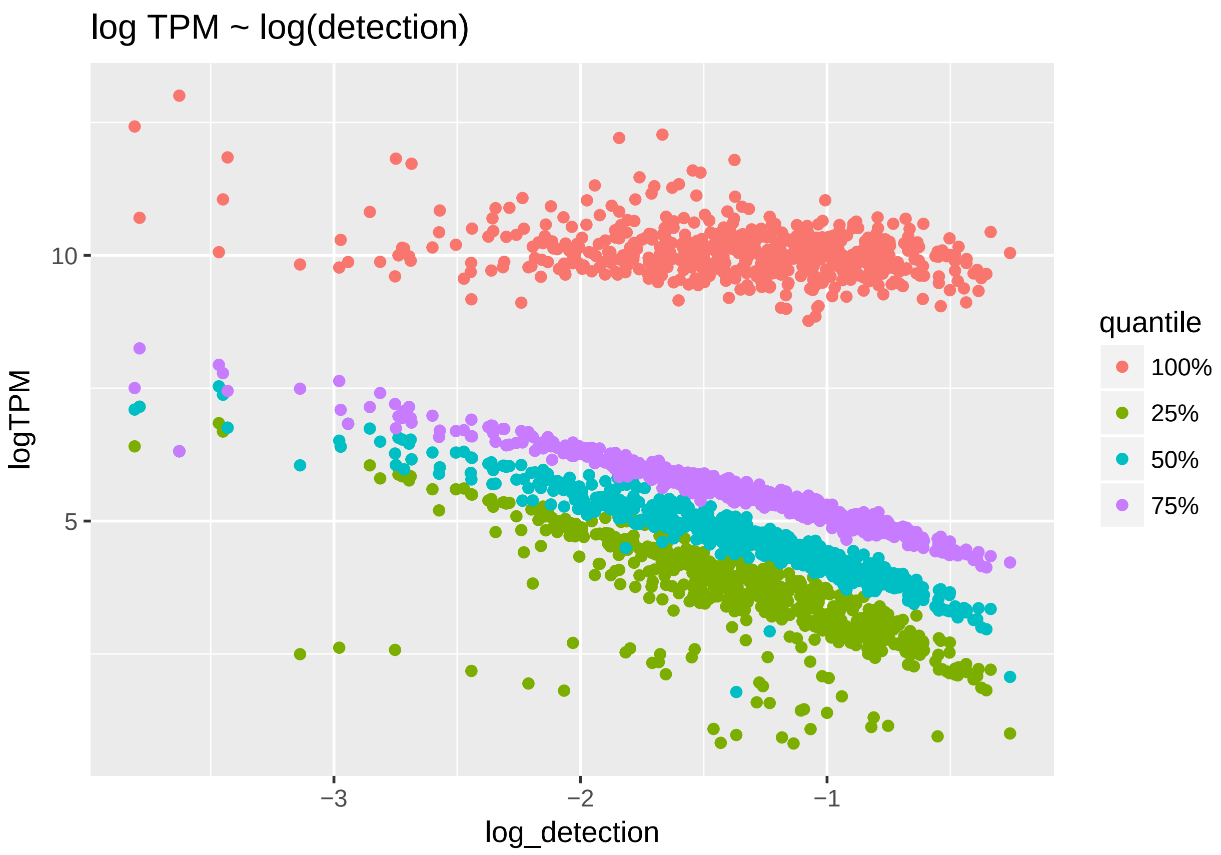
Tae Kim

6/1/2018

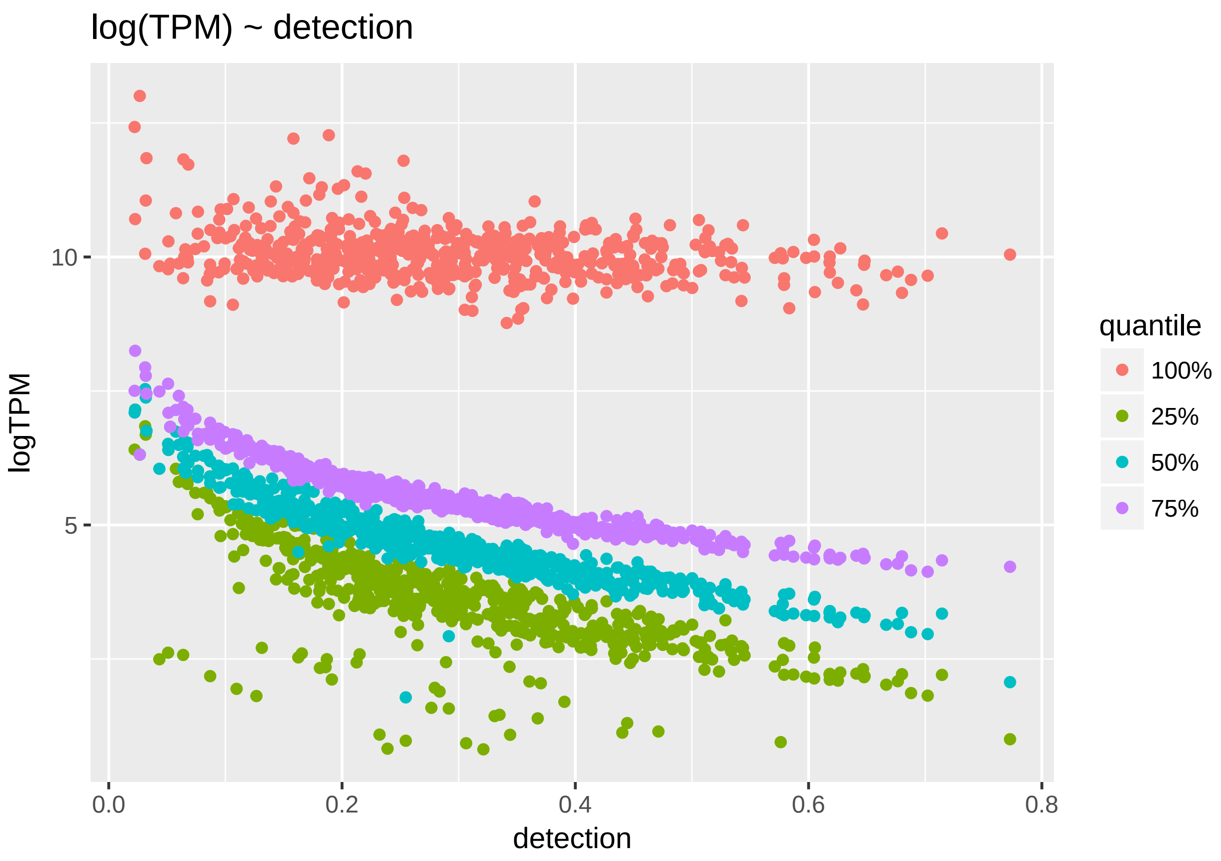
```
exp = read.table('../data/Usoskin_raw_from_excel.txt', header=TRUE)
info = read.table('../data/Usoskin_raw_from_excel_info.txt', header=TRUE)


ind = which(info$level1 %in% c('NF','NP','PEP', 'TH'))
exp = exp[, ind]
info = info[ind, ]
info = apply(info, 2, as.character)
info = as.data.frame(info)
```

```
tpm_orig = exp
tpm = tpm_orig[rowSums(tpm_orig>0)>10, ]
tpm = apply(tpm, 2, as.numeric) #15334 x 622
labels1 = info$level1
labels2 = info$level2
labels3 = info$level3
```

```
ngenes = colSums(tpm>0) # num of genes expressed for each cell
det.rate = ngenes/nrow(tpm) #detection rate
tpm = apply(tpm, 2, as.numeric) #its first pc is not related with det.rate
log.tpm = log(tpm + 1)  #its first pc is linearly related with log(det.rate)

n = ncol(log.tpm)
num.gene = nrow(log.tpm)

quants = apply(log.tpm, 2, function(x) quantile(x[x>0], probs=c(.25, .5, .75, 1)))
type = c(rep('25%',n), rep('50%',n), rep('75%',n), rep('100%',n))
log.quants.df = data.frame(logTPM = c(t(quants)), quantile=type, log_detection = rep(log(det.rate), 4))
ggplot(log.quants.df, aes(x=log_detection, y=logTPM, col=quantile))+
  geom_point()+
  ggtitle('log TPM ~ log(detection)')
```

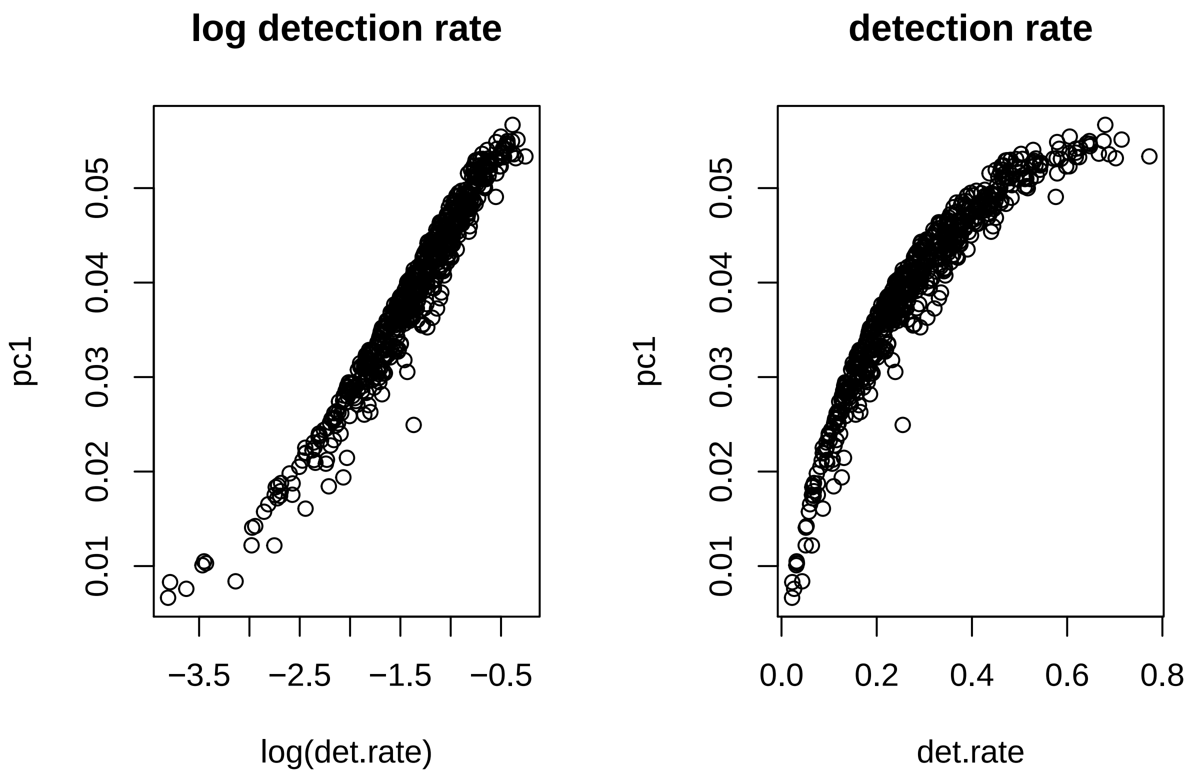


```
quants = apply(log.tpm, 2, function(x) quantile(x[x>0], probs=c(.25, .5, .75, 1)))
type = c(rep('25%',n), rep('50%',n), rep('75%',n), rep('100%',n))
quants.df = data.frame(logTPM = c(t(quants)), quantile=type, detection = rep(det.rate, 4))
ggplot(quants.df, aes(x=detection, y=logTPM, col=quantile))+
  geom_point()+
  ggtitle('log(TPM) ~ detection')
```



The first principal component is correlated with the detection rate

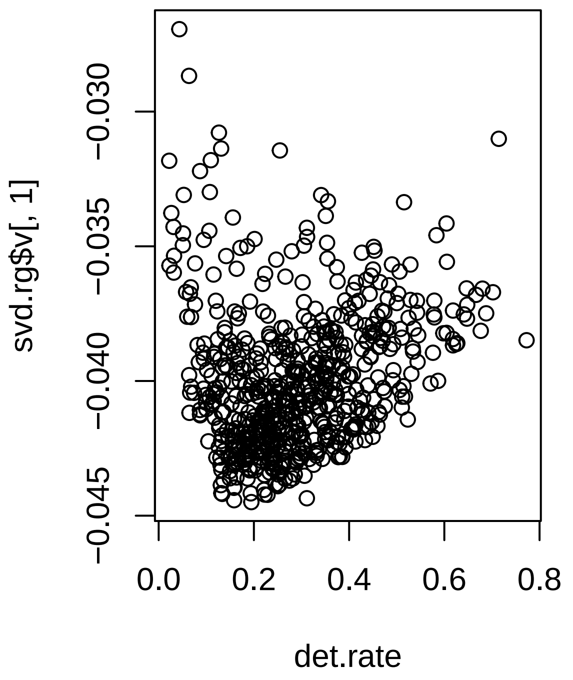
```
svd.base = irlba(log.tpm, 3)
par(mfrow=(c(1,2)))
plot(svd.base$v[,1] ~ log(det.rate), main='log detection rate', ylab = 'pc1')
plot(svd.base$v[,1] ~ det.rate, main='detection rate', ylab = 'pc1')
```

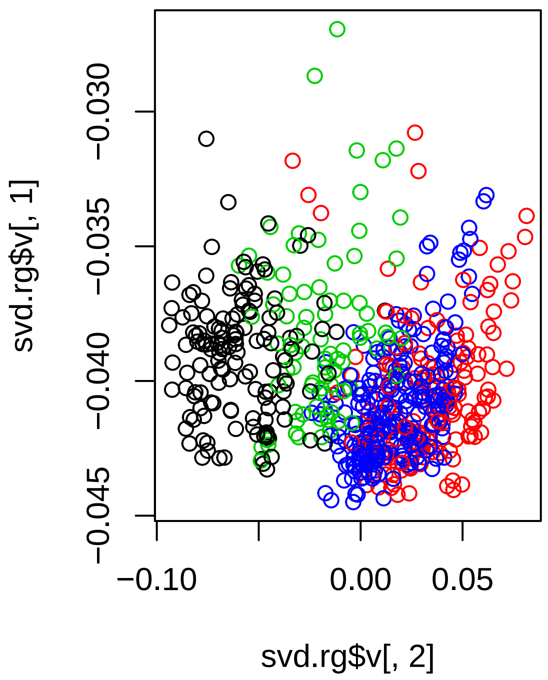


Remove the effects of detection rate

```
log.tpm.rg = log.tpm
x = log(det.rate)-mean(log(det.rate))
log.tpm.rg = t(t(scale(log.tpm)) - 1/sum(x^2) * x %*% (t(x) %*% t(scale(log.tpm))))
gene.var = apply(log.tpm.rg, 1, var)
gene.mean = rowMeans(log.tpm.rg)
```

```
var.genes = which(gene.var>quantile(gene.var,0.3))
log.tpm.rg.filtered = log.tpm.rg[var.genes,]
svd.rg = irlba(log.tpm.rg.filtered, 3)
par(mfrow=c(1,2))
plot(svd.rg$v[,1] ~ det.rate)
plot(svd.rg$v[,1] ~ svd.rg$v[,2], col=labels1)
```

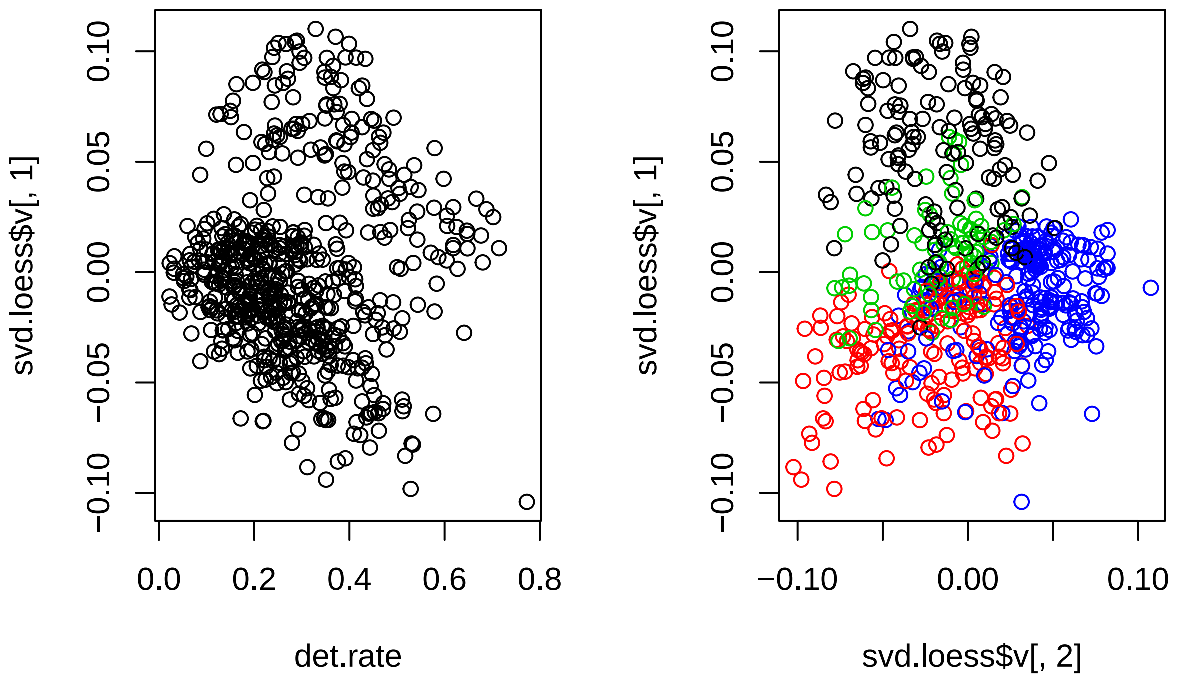




Try Loess as well.

```
log.tpm.loess = log.tpm
for (i in 1:num.gene){
  fit = loess(log.tpm[i,] ~ det.rate)
  log.tpm.loess[i,] = fit$residuals
}
```

```
#select genes
gene.var = apply(log.tpm.loess, 1, var)
gene.mean = rowMeans(log.tpm.loess)
var.genes = which(gene.var>quantile(gene.var, 0.3))
log.tpm.loess.filtered = log.tpm.loess[var.genes, ]
svd.loess = irlba(log.tpm.loess.filtered, 3)
par(mfrow = c(1,2))
plot(svd.loess$v[,1] ~ det.rate)
plot(svd.loess$v[,1] ~ svd.loess$v[,2], col=as.numeric(labels1))
```



```
out.reg = ssl_wrapper(log.tpm.rg.filtered, numClust=length(unique(labels1)), filter=F)
```

```
[1] "constructing kernel.."
[1] "optimizing.."
[1] "network diffusion.."
[1] "dimension reduction.."
```

```
library(SIMLR)
library(pcaReduce)
library(SingleCellExperiment)
library(SC3)
library(gdata)

k = kmeans(t(log.tpm.rg.filtered), 4, nstart=5)$cluster
s = SIMLR(log.tpm.rg.filtered, 4)
```

```
Computing the multiple Kernels.
Performing network diffiusion.
Iteration:  1
Iteration:  2
Iteration:  3
Iteration:  4
Iteration:  5
Iteration:  6
Iteration:  7
Iteration:  8
Iteration:  9
Iteration:  10
Iteration:  11
Iteration:  12
Performing t-SNE.
Epoch: Iteration # 100  error is:  0.1788503
Epoch: Iteration # 200  error is:  0.1690693
```

```
Epoch: Iteration # 300  error is:  0.1670167
Epoch: Iteration # 400  error is:  0.1661941
Epoch: Iteration # 500  error is:  0.1657782
Epoch: Iteration # 600  error is:  0.165301
Epoch: Iteration # 700  error is:  0.165144
Epoch: Iteration # 800  error is:  0.1650399
Epoch: Iteration # 900  error is:  0.1649623
Epoch: Iteration # 1000  error is:  0.1649024
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  12.40347
Epoch: Iteration # 200  error is:  0.2696491
Epoch: Iteration # 300  error is:  0.2643483
Epoch: Iteration # 400  error is:  0.263712
Epoch: Iteration # 500  error is:  0.2633102
Epoch: Iteration # 600  error is:  0.2626216
Epoch: Iteration # 700  error is:  0.2625393
Epoch: Iteration # 800  error is:  0.2624999
Epoch: Iteration # 900  error is:  0.2624796
Epoch: Iteration # 1000  error is:  0.2617713
```

```
p = PCAreduce(t(log.tpm.rg.filtered), 10, 4, 'M')[[1]][,1]
colnames(log.tpm.rg.filtered) = paste0('C',1:ncol(log.tpm.rg.filtered))
rownames(log.tpm.rg.filtered) = paste0('R',1:nrow(log.tpm.rg.filtered))
sce = SingleCellExperiment(
  assays = list(
    counts = exp(as.matrix(log.tpm.rg.filtered))-1,
    logcounts = as.matrix(log.tpm.rg.filtered)
  ),
  colData = colnames(log.tpm.rg.filtered)
)
rowData(sce)$feature_symbol = rownames(log.tpm.rg.filtered)
sc = sc3(sce, ks = 3, biology = FALSE, gene_filter=FALSE)
```

```
sc3_export_results_xls(sc, filename = paste0("sc", ".xls"))
x = read.xls(paste0("sc", ".xls"))
scr = x[,2]

print(paste('kmeans    : ',adj.rand.index(k, as.numeric(labels1))))
```

```
[1] "kmeans    :  0.669148069937991"
```

```
print(paste('SIMLR     : ',adj.rand.index(s$y$cluster, as.numeric(labels1))))
```

```
[1] "SIMLR     :  0.69961216351858"
```

```
print(paste('pcaReduce : ',adj.rand.index(p, as.numeric(labels1))))
```

```
[1] "pcaReduce :  0.56584833749437"
```

```
print(paste('SC3       : ',adj.rand.index(scr, as.numeric(labels1))))
```

```
[1] "SC3       :  0.762746420750613"
```

```
print(paste('SSL       : ',adj.rand.index(out.reg$result, as.numeric(labels1))))
```

```
[1] "SSL       :  0.941764770023187"
```

Without regressing out the detection effect..

```
out = ssl_wrapper(log.tpm, numClust=length(unique(labels1)))
```

```
[1] "constructing kernel.."
[1] "optimizing.."
[1] "network diffusion.."
[1] "dimension reduction.."
```

```
k = kmeans(t(log.tpm), 4, nstart=5)$cluster
s = SIMLR(log.tpm, 4)
```

```
Computing the multiple Kernels.
Performing network diffiusion.
Iteration:  1
Iteration:  2
Iteration:  3
Iteration:  4
Iteration:  5
Iteration:  6
Iteration:  7
Iteration:  8
Iteration:  9
Iteration:  10
Iteration:  11
Performing t-SNE.
Epoch: Iteration # 100  error is:  0.1758487
Epoch: Iteration # 200  error is:  0.1643018
Epoch: Iteration # 300  error is:  0.1605023
Epoch: Iteration # 400  error is:  0.158882
Epoch: Iteration # 500  error is:  0.1580454
Epoch: Iteration # 600  error is:  0.1575224
Epoch: Iteration # 700  error is:  0.1571528
Epoch: Iteration # 800  error is:  0.1568489
Epoch: Iteration # 900  error is:  0.1566334
Epoch: Iteration # 1000  error is:  0.1564639
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  11.49606
Epoch: Iteration # 200  error is:  0.2575464
Epoch: Iteration # 300  error is:  0.2474824
Epoch: Iteration # 400  error is:  0.2455118
Epoch: Iteration # 500  error is:  0.2447283
Epoch: Iteration # 600  error is:  0.2442843
Epoch: Iteration # 700  error is:  0.2439789
Epoch: Iteration # 800  error is:  0.2437631
Epoch: Iteration # 900  error is:  0.2435952
Epoch: Iteration # 1000  error is:  0.2434669
```

```
p = PCAreduce(t(log.tpm), 10, 4, 'M')[[1]][,1]
colnames(log.tpm) = paste0('C',1:ncol(log.tpm))
rownames(log.tpm) = paste0('R',1:nrow(log.tpm))
sce = SingleCellExperiment(
  assays = list(
    counts = exp(as.matrix(log.tpm))-1,
    logcounts = as.matrix(log.tpm)
```

```r
  ),
  colData = colnames(log.tpm)
)
rowData(sce)$feature_symbol = rownames(log.tpm)
sc = sc3(sce, ks = 3, biology = FALSE, gene_filter=FALSE)
```

```r
sc3_export_results_xls(sc, filename = paste0("sc", ".xls"))
x = read.xls(paste0("sc", ".xls"))
scr = x[,2]

print(paste('kmeans    : ',adj.rand.index(k, as.numeric(labels1))))
```

```
[1] "kmeans    :  0.231900213263911"
```

```r
print(paste('SIMLR     : ',adj.rand.index(s$y$cluster, as.numeric(labels1))))
```

```
[1] "SIMLR     :  0.658038109785475"
```

```r
print(paste('pcaReduce : ',adj.rand.index(p, as.numeric(labels1))))
```

```
[1] "pcaReduce :  0.344419272840963"
```

```r
print(paste('SC3       : ',adj.rand.index(scr, as.numeric(labels1))))
```

```
[1] "SC3       :  0.772889722316618"
```

```r
print(paste('SSL       : ',adj.rand.index(out$result, as.numeric(labels1))))
```

```
[1] "SSL       :  0.643621259488143"
```