Pollen

Tae Kim

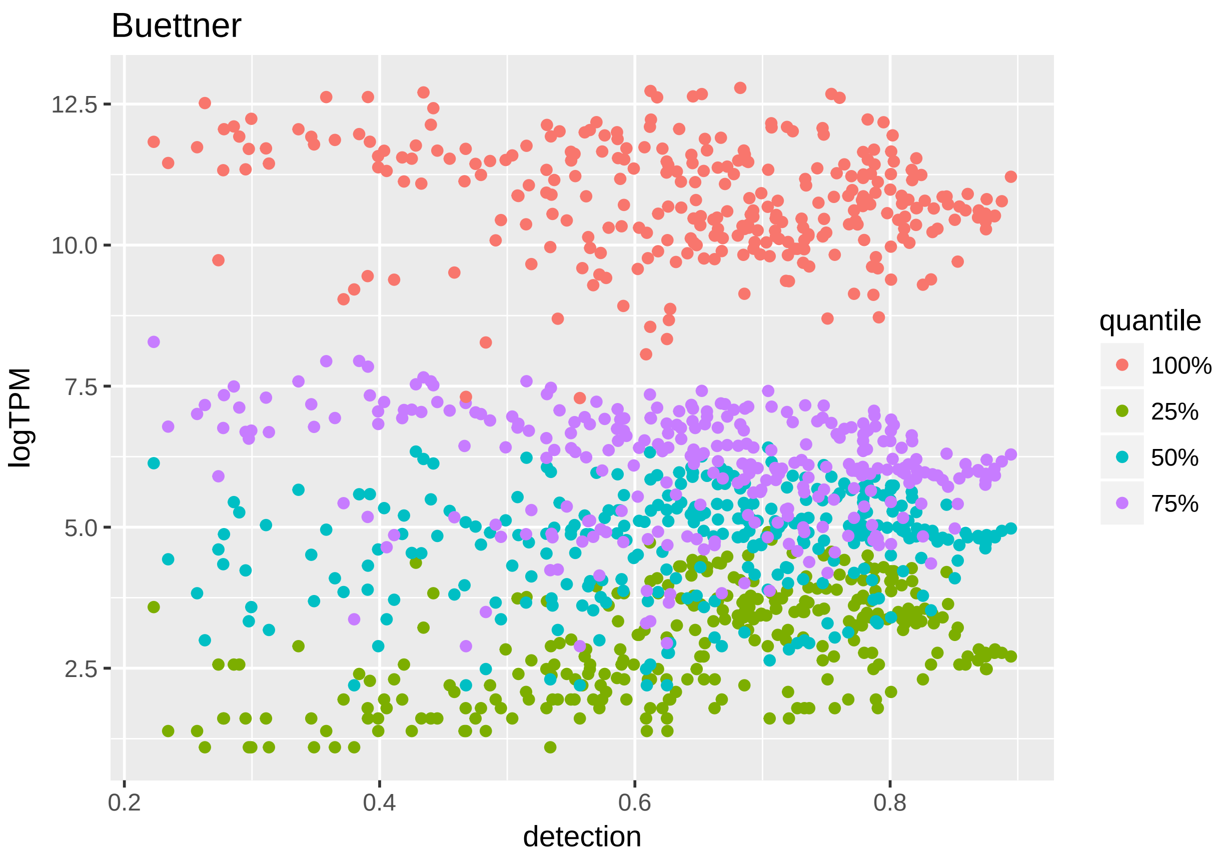
6/1/2018

```
tpm_orig = read.csv('../../../benchmark/pcaReduce-master/Pollen2014.txt', header=TRUE)
label = as.factor(sapply(strsplit(colnames(tpm_orig), '_'), function(x) x[2]))
```

```
tpm = tpm_orig[rowSums(tpm_orig>0)>3, ]
print(dim(tpm))
```

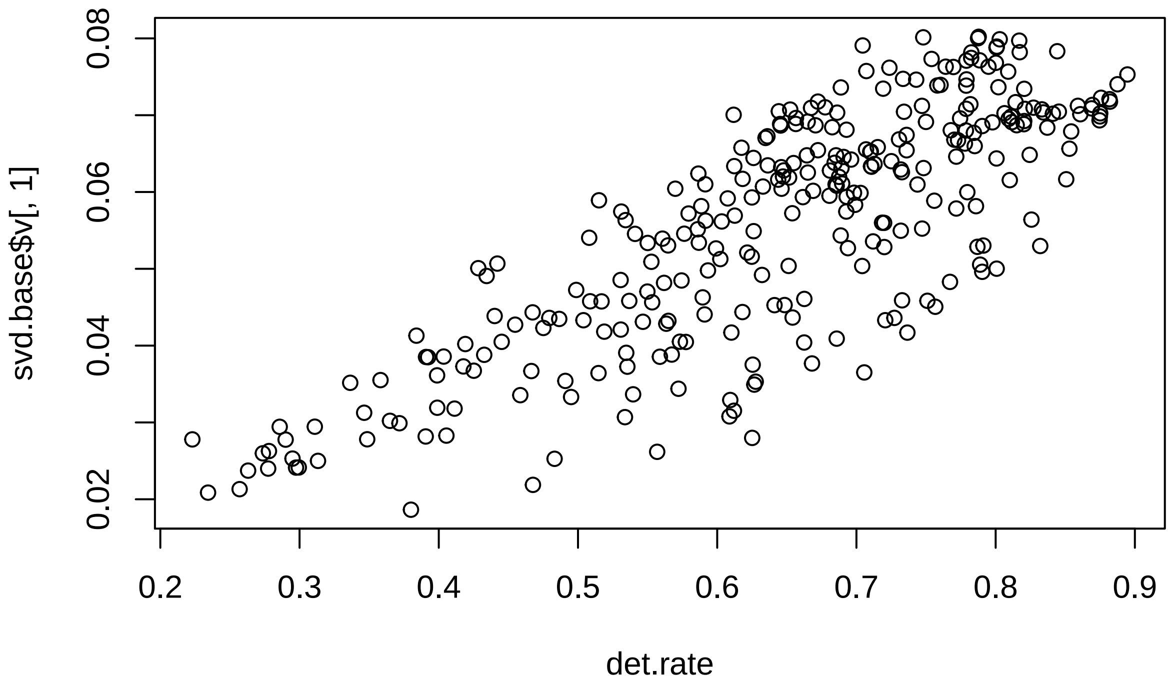
```
[1] 8686  300
```

```
ngenes = colSums(tpm>0)
det.rate = ngenes/nrow(tpm)
log.tpm = log(tpm + 1)

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))
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('Buettner')
```



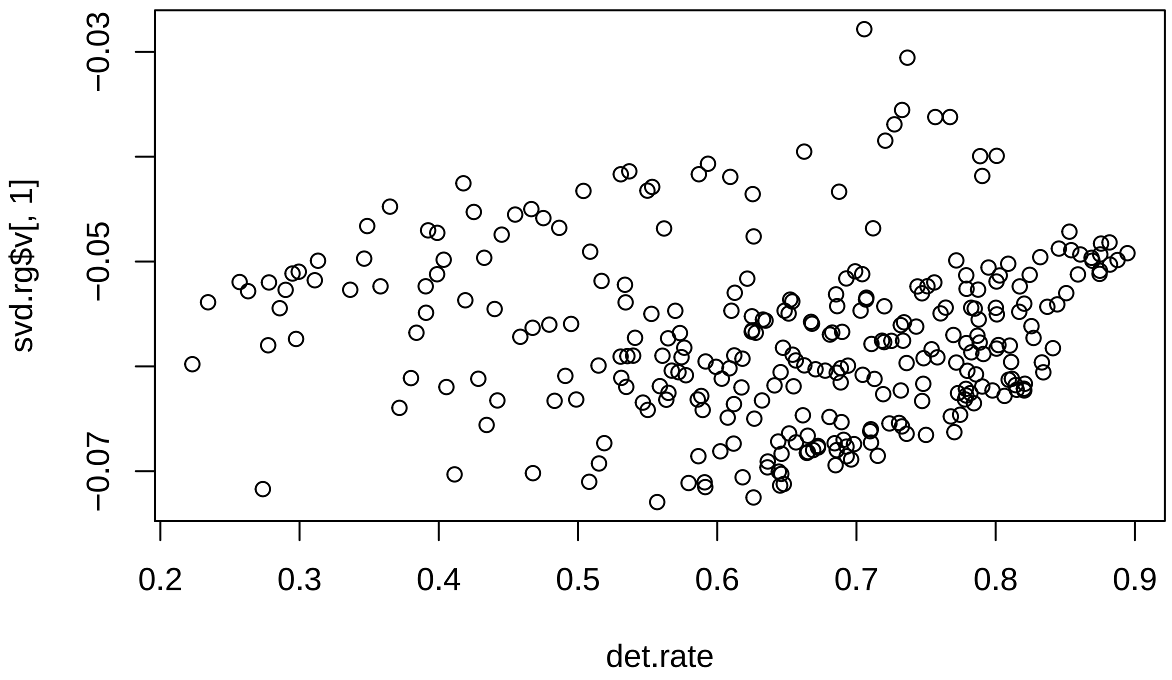
The first principal component is correlated with the detection rate

```
svd.base = irlba(as.matrix(log.tpm), 3)
plot(svd.base$v[,1] ~ det.rate)
```



Remove the effects of detection rate

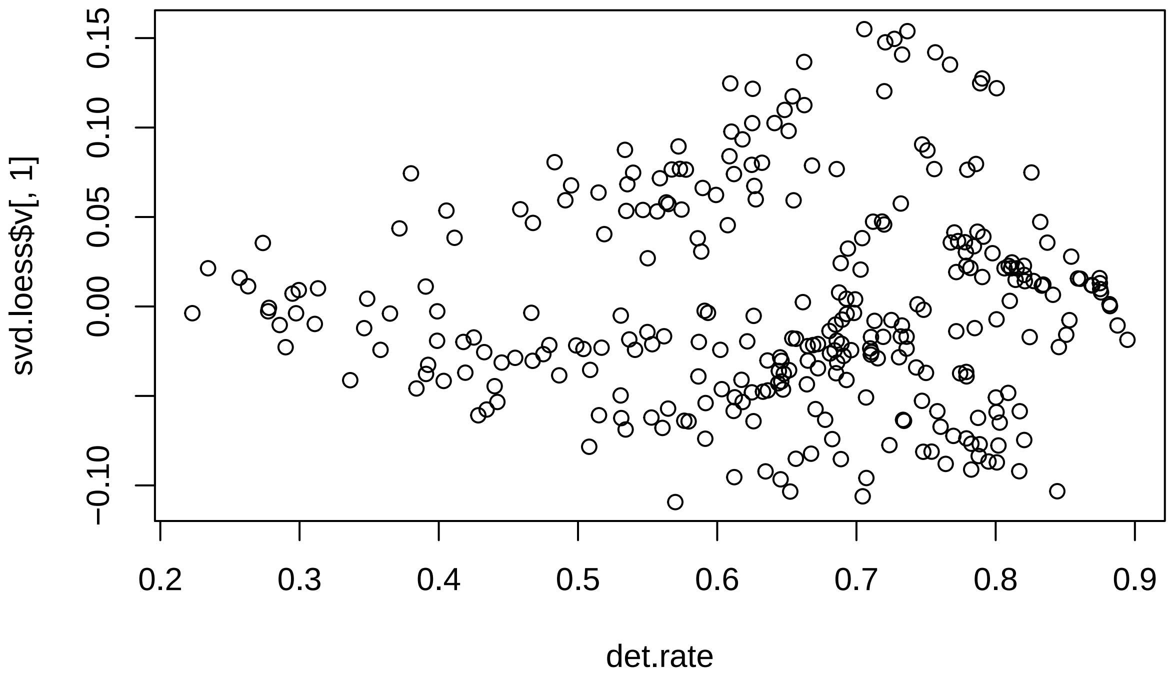
```
x = det.rate - mean(det.rate)
log.tpm.rg = t(t(scale(log.tpm)) - x %*% t(x) %*% t(scale(log.tpm))/sum(x^2))

gene.var = apply(log.tpm.rg, 1, var)
gene.mean = rowMeans(log.tpm.rg)
var.genes = names(gene.var[gene.var>quantile(gene.var, 0.3)])
log.tpm.rg.filtered = log.tpm.rg[var.genes,]
svd.rg = irlba(log.tpm.rg.filtered, 2)
plot(svd.rg$v[,1] ~ det.rate)
```



Trying loess..

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

```
#select genes
gene.var = apply(log.tpm.loess, 1, var)
gene.mean = rowMeans(log.tpm.loess)
var.genes = names(gene.var[gene.var>quantile(gene.var, 0.3)])
log.tpm.loess.filtered = log.tpm.loess[var.genes, ]
svd.loess = irlba(log.tpm.loess.filtered, 2)
plot(svd.loess$v[,1] ~ det.rate)
```



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

out.rg = ssl_wrapper(log.tpm.rg.filtered, numClust=11)
```

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

```
k = kmeans(t(log.tpm.rg.filtered), 11, nstart=5)$cluster
s = SIMLR(log.tpm.rg.filtered, 11)
```

```
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
Performing t-SNE.
Epoch: Iteration # 100  error is:  0.05762765
Epoch: Iteration # 200  error is:  0.01704161
Epoch: Iteration # 300  error is:  0.01604606
Epoch: Iteration # 400  error is:  0.01535186
```

```
Epoch: Iteration # 500  error is:  0.01481403
Epoch: Iteration # 600  error is:  0.0143792
Epoch: Iteration # 700  error is:  0.01402133
Epoch: Iteration # 800  error is:  0.01370993
Epoch: Iteration # 900  error is:  0.01342935
Epoch: Iteration # 1000  error is:  0.01318106
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  11.55392
Epoch: Iteration # 200  error is:  0.1216479
Epoch: Iteration # 300  error is:  0.06373301
Epoch: Iteration # 400  error is:  0.06116588
Epoch: Iteration # 500  error is:  0.05941155
Epoch: Iteration # 600  error is:  0.05809447
Epoch: Iteration # 700  error is:  0.05703761
Epoch: Iteration # 800  error is:  0.05619227
Epoch: Iteration # 900  error is:  0.05548752
Epoch: Iteration # 1000  error is:  0.05488762
```

```r
p = PCAreduce(t(log.tpm.rg.filtered), 10, 11, '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 = 11, 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(label))))
```

```
[1] "kmeans    :  0.905142667562919"
```

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

```
[1] "SIMLR     :  0.852292475049147"
```

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

```
[1] "pcaReduce :  0.573906826774459"
```

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

```
[1] "SC3       :  0.8340854768695"
```

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

```
[1] "SSL       :  0.775578772030702"
```

Without regressing out the detection effect..

```r
out = ssl_wrapper(log.tpm, numClust=11)
```

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

```r
k = kmeans(t(log.tpm), 11, nstart=5)$cluster
s = SIMLR(log.tpm, 11)
```

```
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
Performing t-SNE.
Epoch: Iteration # 100  error is:  0.0470446
Epoch: Iteration # 200  error is:  0.03538361
Epoch: Iteration # 300  error is:  0.0321973
Epoch: Iteration # 400  error is:  0.0302687
Epoch: Iteration # 500  error is:  0.02896625
Epoch: Iteration # 600  error is:  0.0279508
Epoch: Iteration # 700  error is:  0.02714153
Epoch: Iteration # 800  error is:  0.02648243
Epoch: Iteration # 900  error is:  0.02592393
Epoch: Iteration # 1000  error is:  0.02545169
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  11.12686
Epoch: Iteration # 200  error is:  0.07767143
Epoch: Iteration # 300  error is:  0.06376837
Epoch: Iteration # 400  error is:  0.06138573
Epoch: Iteration # 500  error is:  0.05970046
Epoch: Iteration # 600  error is:  0.05845306
Epoch: Iteration # 700  error is:  0.057464
Epoch: Iteration # 800  error is:  0.05666078
Epoch: Iteration # 900  error is:  0.05600369
Epoch: Iteration # 1000  error is:  0.05544603
```

```r
p = PCAreduce(t(log.tpm), 10, 11, '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)
  ),
  colData = colnames(log.tpm)
)
rowData(sce)$feature_symbol = rownames(log.tpm)
sc = sc3(sce, ks = 11, 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(label))))
```

```
[1] "kmeans    :  0.793891629960835"
```

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

```
[1] "SIMLR     :  0.67823051042753"
```

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

```
[1] "pcaReduce :  0.866864795535503"
```

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

```
[1] "SC3       :  0.934127039090031"
```

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

```
[1] "SSL       :  0.793118564179666"
```