Buettner

Tae Kim

6/1/2018

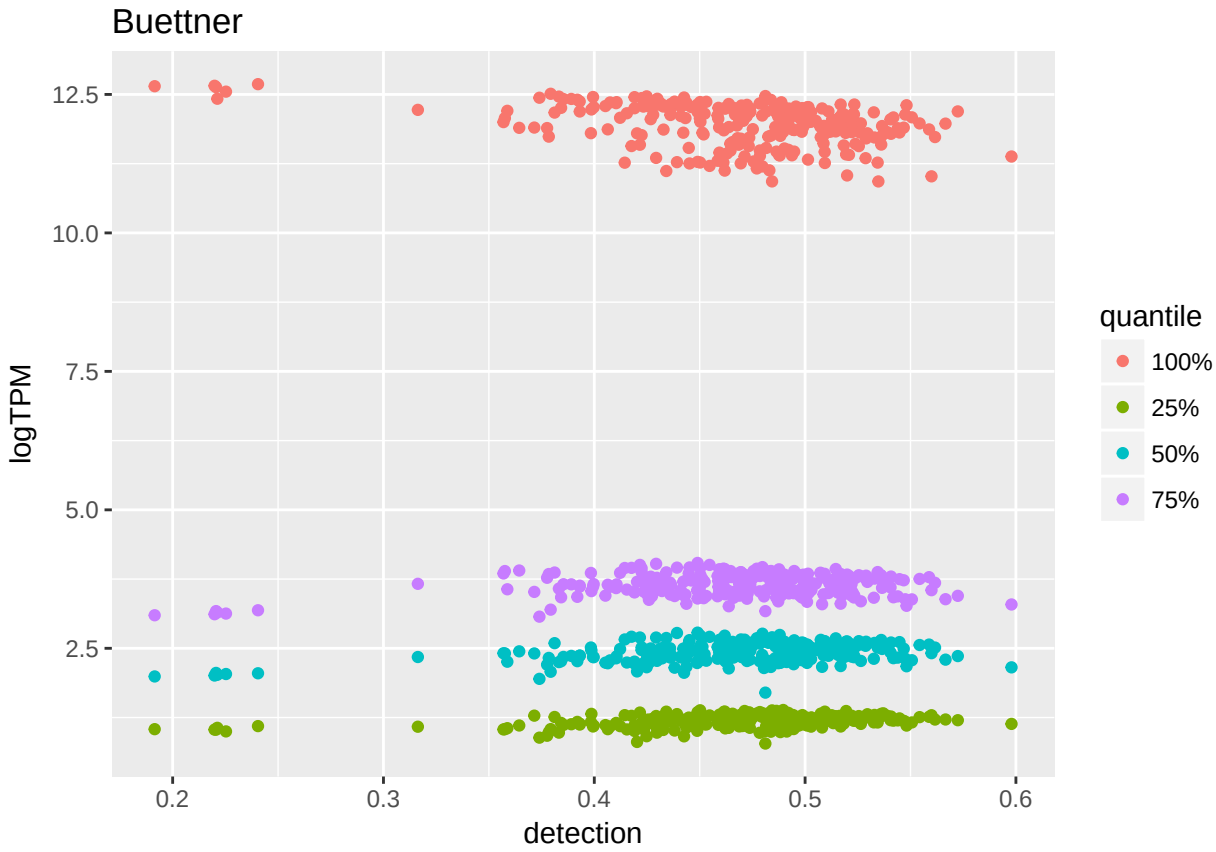
```
b = readRDS('../data/EMTAB2805PMID25599176_Buettner2015.rds')
genes = experiments(b)
tpm_orig = assays(genes)[['gene']]
label = as.factor(b$cell_cycle_stage)
rm(b)
rm(genes)
gc()
```

```
           used  (Mb) gc trigger   (Mb) limit (Mb)  max used   (Mb)
Ncells  6406465 342.2   12432902  664.0         NA   8813222  470.7
Vcells 24016426 183.3  236035886 1800.9      16384 275044364 2098.5
```

```
tpm = tpm_orig[rowSums(tpm_orig>0)>3, ]
remove_cells = which(colSums(tpm_orig>0)/nrow(tpm_orig)<0.1)
tpm= tpm[, -remove_cells]
label = label[-remove_cells]
print(dim(tpm))
```

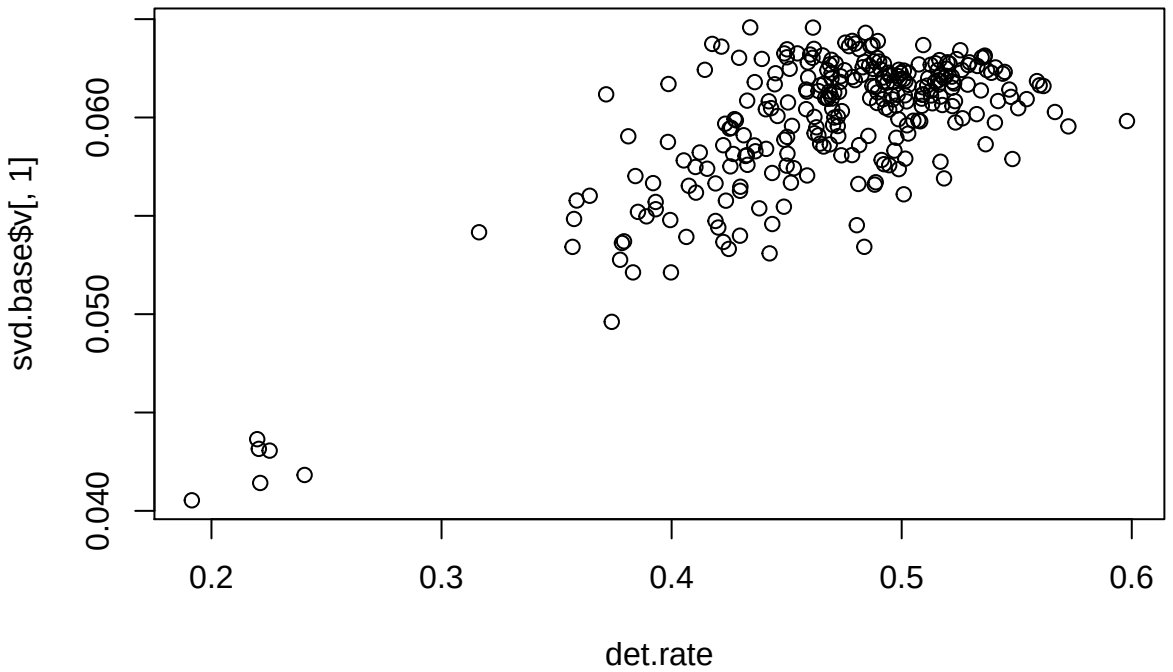
```
[1] 25873   279
```

```
ngenes = colSums(tpm>0)
det.rate = ngenes/nrow(tpm)
log.tpm = log(tpm + 0.1)

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')
```



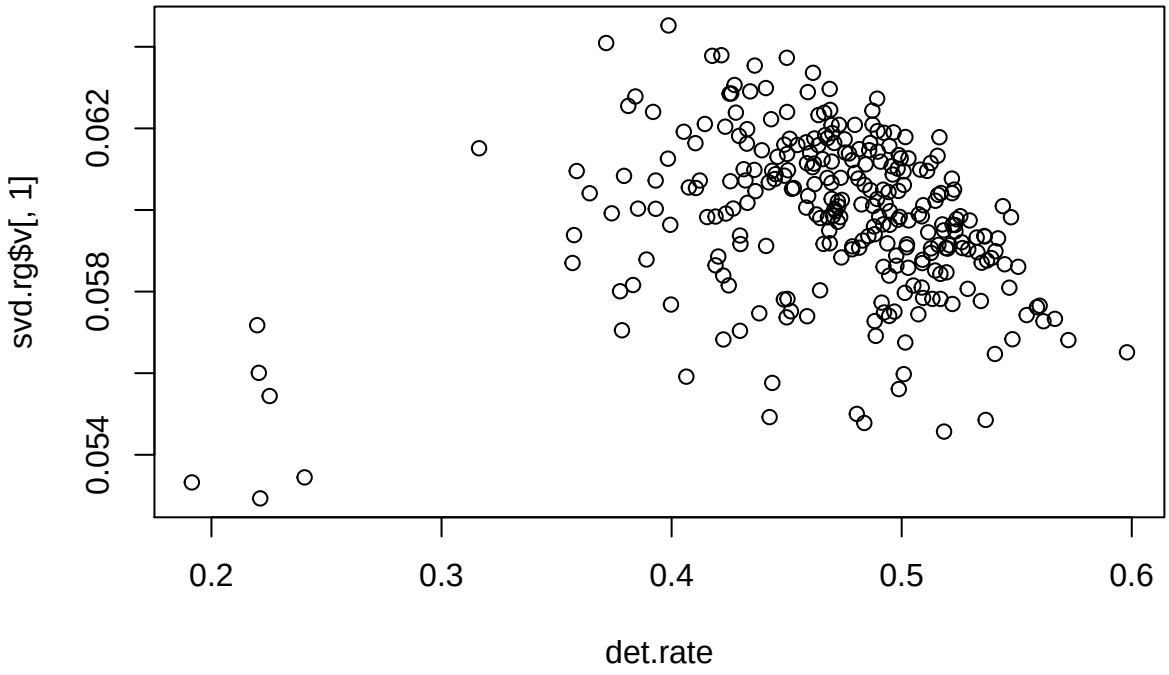
The first principal component is correlated with the detection rate

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



Remove the effects of detection rate

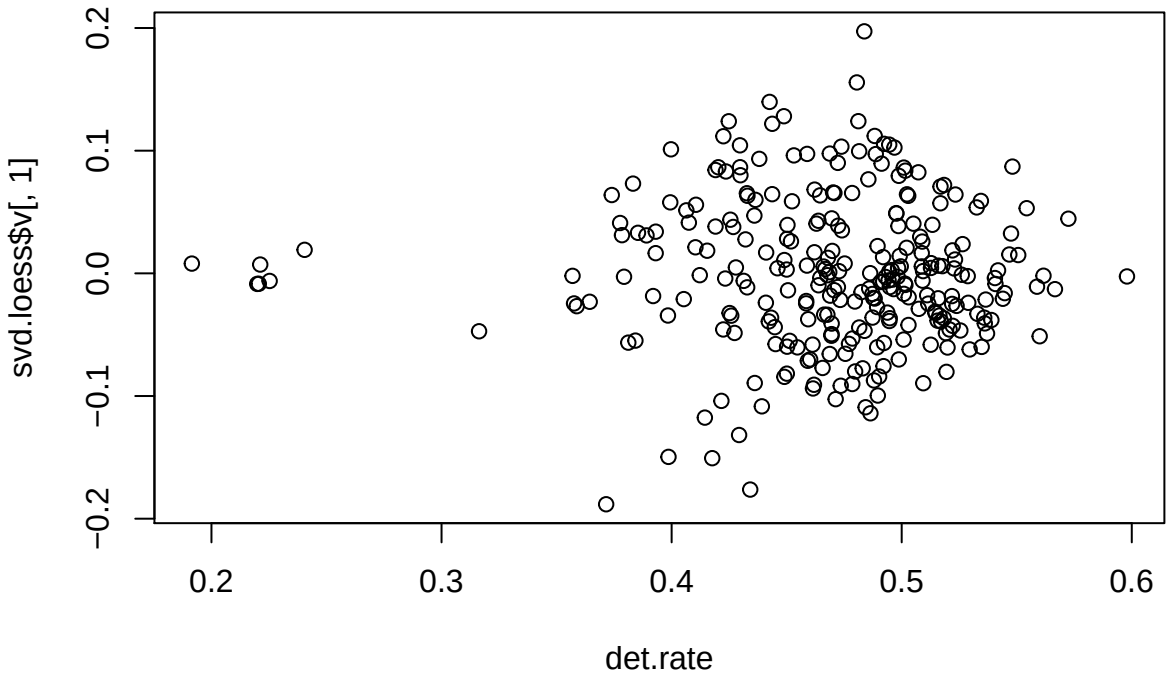
```
x = det.rate - mean(det.rate)
log.tpm.rg = t(t(scale(log.tpm)) - x  %*% t(x) %*% t(scale(log.tpm))/sum(x^2))
gene.var = apply(log.tpm.rg, 1, var)
var.genes = names(gene.var[gene.var>quantile(gene.var, 0.3)])
log.tpm.rg.filtered = log.tpm.rg[var.genes,]
svd.rg = irlba(log.tpm.rg.filtered, 2)
plot(svd.rg$v[,1] ~ det.rate)
```



There are still the detection rate effect in the first principal component. Try Loess.

```
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>1)
log.tpm.loess.filtered = log.tpm.loess[var.genes, ]
svd.loess = irlba(log.tpm.loess.filtered, 2)
plot(svd.loess$v[,1] ~ det.rate)
```




```
out_loess = ssl_wrapper(log.tpm.loess.filtered, numClust=3)
```

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

```
out_reg = ssl_wrapper(log.tpm.rg.filtered, numClust=3)
```

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

```
k = kmeans(t(log.tpm.rg.filtered), 3, nstart=5)$cluster
s = SIMLR(log.tpm.rg.filtered, 3)
```

```
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:  0.190581
Epoch: Iteration # 200  error is:  0.2291951
Epoch: Iteration # 300  error is:  0.2481565
```

```
Epoch: Iteration # 400  error is:  0.2424379
Epoch: Iteration # 500  error is:  0.2268119
Epoch: Iteration # 600  error is:  0.1721993
Epoch: Iteration # 700  error is:  0.1611862
Epoch: Iteration # 800  error is:  0.1634423
Epoch: Iteration # 900  error is:  0.1646566
Epoch: Iteration # 1000  error is:  0.1994541
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  13.32034
Epoch: Iteration # 200  error is:  0.543669
Epoch: Iteration # 300  error is:  0.3868201
Epoch: Iteration # 400  error is:  0.3727963
Epoch: Iteration # 500  error is:  0.3426174
Epoch: Iteration # 600  error is:  0.3255775
Epoch: Iteration # 700  error is:  0.3620333
Epoch: Iteration # 800  error is:  0.3668882
Epoch: Iteration # 900  error is:  0.3697356
Epoch: Iteration # 1000  error is:  0.3753321
```

```
p = PCAreduce(t(log.tpm.rg.filtered), 10, 3, '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(label))))
```

```
[1] "kmeans    :  0.408452492419273"
```

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

```
[1] "SIMLR     :  0.285178676553722"
```

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

```
[1] "pcaReduce :  0.515716645600848"
```

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

```
[1] "SC3       :  0.481987739004395"
```

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

```
[1] "SSL       :  0.64731974065531"
```

```
out_orig = ssl_wrapper(log.tpm, numClust=3)
```

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

```
k = kmeans(t(log.tpm), 3, nstart=5)$cluster
s = SIMLR(log.tpm, 3)
```

```
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:  0.1374671
Epoch: Iteration # 200  error is:  0.1308583
Epoch: Iteration # 300  error is:  0.130126
Epoch: Iteration # 400  error is:  0.1299273
Epoch: Iteration # 500  error is:  0.1298
Epoch: Iteration # 600  error is:  0.1296911
Epoch: Iteration # 700  error is:  0.1295994
Epoch: Iteration # 800  error is:  0.129528
Epoch: Iteration # 900  error is:  0.129465
Epoch: Iteration # 1000  error is:  0.1294194
Performing Kmeans.
Performing t-SNE.
Epoch: Iteration # 100  error is:  11.97315
Epoch: Iteration # 200  error is:  0.3874081
Epoch: Iteration # 300  error is:  0.291139
Epoch: Iteration # 400  error is:  0.2512413
Epoch: Iteration # 500  error is:  0.2601702
Epoch: Iteration # 600  error is:  0.2806714
Epoch: Iteration # 700  error is:  0.2636508
Epoch: Iteration # 800  error is:  0.2851401
Epoch: Iteration # 900  error is:  0.2734582
Epoch: Iteration # 1000  error is:  0.2819064
```

```
p = PCAreduce(t(log.tpm), 10, 3, '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 = 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(label))))
```

```
[1] "kmeans    :  0.0518156566606963"
```

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

```
[1] "SIMLR     :  0.299251922192462"
```

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

```
[1] "pcaReduce :  0.220107016733817"
```

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

```
[1] "SC3       :  0.49698778772153"
```

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

```
[1] "SSL       :  0.408740795494007"
```