Response to McDavid et al.

Florian Buettner, John C. Marioni and Oliver Stegle

Contents


```
##
## groupGOTerms:    GOBPTerm, GOMFTerm, GOCCTerm environments built.
```

Loading the data

We first follow McDavid et al. and load all the required data, including corrected and uncorrected T-cell expression data, the results of the clustering of the correced T-cell data and the cell-cycle genes.

```
## Supplementary Data 1, sheet 1
T_cell_raw <- fread("data/T_cell_uncorrected.csv")
## sheet 2
T_cell_corrected <- fread("data/T_cell_corrected.csv")
setnames(T_cell_raw, "V1", "cell_id")
setnames(T_cell_corrected, "V1", "cell_id")
## Supplementary Data 1, sheet 3
cluster <- fread("data/T_cell_cluster.csv")
setnames(cluster, 'Gata3HighCLuster', 'clusterid')
cluster <- cluster[,clusterid:=factor(clusterid)]

T_cell_matrix <- as.matrix(T_cell_raw[,-1,with=FALSE])
T_cell_matrix_corrected <- as.matrix(T_cell_corrected[,-1,with=FALSE])
# Remove duplicated gene names (only 2)
T_cell_matrix <- T_cell_matrix[,unique(colnames(T_cell_matrix))]
T_cell_matrix_corrected <- T_cell_matrix_corrected[,unique(colnames(T_cell_matrix_corrected))]
geneID = colnames(T_cell_matrix_corrected)
stopifnot(all(colnames(T_cell_matrix_corrected)==colnames(T_cell_matrix)))

TcellCdat <- data.frame(cluster)
```

Relationship between scLVM factor and different size factors

In order to explore how well the inferred factor by scLVM tracks cell size and other features, we considered alternative approaches for estimating size factors. These include BASiCS, a Bayesian method which infers separate size factors corresponding to cell size and technical variation as well as the standard DESeq size factors.

```
#1. laod raw read counts
data(data_Tcells)
geneTypes <- factor( c( ENSM="ENSM", ERCC="ERCC" )[
substr( rownames(dataMouse), 1, 4 ) ] )
countsMmus <- dataMouse[ which( geneTypes=="ENSM" ), ]
sym_names = getSymbols(row.names(countsMmus))
countsERCC <- dataMouse[ which( geneTypes=="ERCC" ), ]

#2. calculate DESeq size factors for counts
sfERCC <- estimateSizeFactorsForMatrix( countsERCC )
sfEndo = estimateSizeFactorsForMatrix( countsMmus )

nCountsERCC <- t( t(countsERCC) / sfERCC )
nCountsMmus <- t( t(countsMmus) / sfERCC )
```

Next, we use the raw data to fit a variety of size factors, as well the scLVM factor. First, let's fit the scLVM factor.

```
Y = t(log10(nCountsMmus+1)) #normalised transformed
sclvm = new("scLVM")
sclvm = init(sclvm,Y=Y,tech_noise = NULL)

#get cell cycle genes from GO
ens_ids_cc <- getEnsembl('GO:0007049')
CellCycle = fitFactor(sclvm,geneSet = ens_ids_cc,k=1)

#Get cell-cycle factor
scLVM.CellCycle = CellCycle$X
```

We have already calculated the DESeq size facotrs, now we will compute the BASiCS size factors. As inference in BASiCS is done via MCMC, computation can take a long time. Fo rconvenience we have stored the results which were generated as follows.

```
 is_expressed = rowSums(countsMmus)>0
 is_expressedERCC = rowSums(countsERCC)>0

 Counts = as.matrix(rbind(countsMmus[is_expressed,],countsERCC[is_expressedERCC,]))
 Tech = c(rep(FALSE,sum(is_expressed)),rep(TRUE,sum(is_expressedERCC)))
 SpikeInput = (apply(countsERCC[is_expressedERCC,],1,mean))

 Filter = BASiCS_Filter(Counts, Tech, SpikeInput,
                        MinTotalCountsPerCell = 2, MinTotalCountsPerGene = 2,
                        MinCellsWithExpression = 2, MinAvCountsPerCellsWithExpression = 2)

 FilterData = newBASiCS_Data(Filter$Counts, Filter$Tech, Filter$SpikeInput)

#not run
# MCMC_Output <- BASiCS_MCMC(FilterData, N = 20000, Thin = 10, Burn = 10000, StoreChains = T,
#                            StoreDir = getwd(), RunName = "Tcells")


load('im_scBasics2708.rda')
#load('../data/res_BASiCS.rda')
```

```
MCMC_Summary <- Summary(MCMC_Output)
Basics.phi = MCMC_Summary@phi[,1] #designed to capture cell size
Basics.s = MCMC_Summary@s[,1] #designed to capture technical variation
```

Finally, we calculate standard library suze by summing up all read counts in a cell and look at the correlation between the various soze factors. This revealed that the scLVM factor is moderately correlated with cell size factors (max correlation R2 = 0.74 with BASiCS).

```
#library size
libsize = apply((countsMmus),2,sum)
libsizeERCC = apply((countsERCC),2,sum)


#correlation between the size factors
corMat = cor(cbind(sfEndo, sfERCC, libsize, libsizeERCC, scLVM.CellCycle, Basics.phi, Basics.s))
```

```
heatmap.2(corMat^2, trace='none', margins=c(10,10))
```

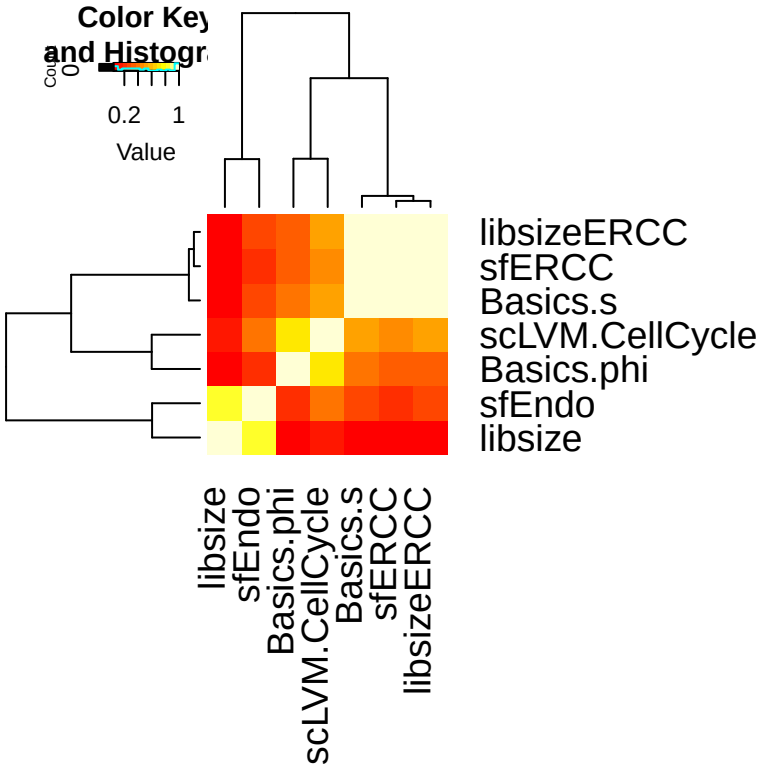


Figure 1: Correlation between scLVM facotr and various size factors

Remark on the analysis presented by McDavid et al.

The above observation is not surprising since cell size and cell cycle stage are clearly related. McDavid et al. used an unconventional approach to compute a "geometric size factor" that uses only a subset of genes and computes the size factor on the normalized log-scale. They observe that this geometric size factor is strongly correlated (R2=0.9) with the scLVM factor and therefore hypothesise that the scLVM factor captures

information only relating to cell size. However, the much lower correlation of the scLVM factor with the independently computed normalization in BASiCS, which is explicitly designed to account for variation in cell size, as well as other size factors estimates in common usage, suggests that the scLVM factor does capture information that is independent of cell size.

Gene Set Enrichment Analysis using Corrected Data

A concern of McDavid et al. is that our analytical strategy does not effectively remove variation due to the cell cycle in the T-cell data. Arguing against this interpretation, in our published manuscript we performed extensive analyses that demonstrated that the residual variability in the cell cycle corrected T-cell data is dominated by factors attributable to T-cell differentiation. These observations are apparently contradicted by the results of the gene set enrichment analysis reported by McDavid et al., who found that the overwhelming majority of enriched modules were cell-cycle related processes. We believe this result can be explained by the analytical methods McDavid et al. applied. Rather than performing a standard GO enrichment that exploits a pre-defined set of differentially expressed genes, McDavid et al. used CAMERA, a competitive gene set test enrichment approach that penalizes inter-gene correlation. CAMERA was developed for microarray data where a high level of inter-gene correlation within the test set of genes (after controlling for the treatment effect) is potentially a confounding factor that can lead to type I errors. In other words, CAMERA assumes that, after correcting for between group mean effects, any correlation remaining between genes is not biologically relevant.

However, in the context of the T-cell differentiation experiment, where the 81 cells were sampled from a differentiation trajectory, we expect that even after allocating cells into two groups, substantial inter-gene correlation within each group will remain and be biologically relevant. More precisely, the two groups we identify contain early and late differentiating cells. Within each of these groups, however, we still expect a gradient of cells: for example, in the "less differentiated group" we will still be able to rank cells along a developmental trajectory.

In order to illustrate this effect, we first compare the inter-gene correlation for genes annotated to moduels identified by us to those identified by McDavid et al. We follow McDavid et al. in using the Broad Institute's "Reactome" module and first run CAMERA.

```
eSet <- ExpressionSet(t(T_cell_matrix_corrected))
pData(eSet) <- TcellCdat

c2_set <- getGmt("data/c2.cp.reactome.v4.0.symbols.gmt")
gene_ids <- geneIds(c2_set)

## # Camera requires gene-indices
design <- model.matrix(~clusterid, eSet)
sets_indices <- ids2indices(gene_ids, toupper(rownames(eSet)))
res <- camera(eSet, sets_indices, design=design, use.ranks = F, sort=F)
resCAMERASorted = res[order(res[,5]),]
resCAMERASorted$set <- row.names(resCAMERASorted)
head(res[order(res[,5]),])
```

```
##                                                                    NGenes
## REACTOME_MITOTIC_PROMETAPHASE                                          66
## REACTOME_E2F_MEDIATED_REGULATION_OF_DNA_REPLICATION                    27
## REACTOME_DEPOSITION_OF_NEW_CENPA_CONTAINING_NUCLEOSOMES_AT_THE_CENTROMERE     19
## REACTOME_G2_M_CHECKPOINTS                                              36
## REACTOME_G1_S_SPECIFIC_TRANSCRIPTION                                   14
## REACTOME_CELL_CYCLE                                                   257
##                                                                   Correlation
```

```
## REACTOME_MITOTIC_PROMETAPHASE                                                0.013189599
## REACTOME_E2F_MEDIATED_REGULATION_OF_DNA_REPLICATION                          0.009275600
## REACTOME_DEPOSITION_OF_NEW_CENPA_CONTAINING_NUCLEOSOMES_AT_THE_CENTROMERE -0.008336669
## REACTOME_G2_M_CHECKPOINTS                                                    0.019916023
## REACTOME_G1_S_SPECIFIC_TRANSCRIPTION                                         0.043079422
## REACTOME_CELL_CYCLE                                                          0.007355654
##                                                                            Direction
## REACTOME_MITOTIC_PROMETAPHASE                                                   Down
## REACTOME_E2F_MEDIATED_REGULATION_OF_DNA_REPLICATION                             Down
## REACTOME_DEPOSITION_OF_NEW_CENPA_CONTAINING_NUCLEOSOMES_AT_THE_CENTROMERE       Down
## REACTOME_G2_M_CHECKPOINTS                                                       Down
## REACTOME_G1_S_SPECIFIC_TRANSCRIPTION                                            Down
## REACTOME_CELL_CYCLE                                                             Down
##                                                                             PValue
## REACTOME_MITOTIC_PROMETAPHASE                                          5.451636e-09
## REACTOME_E2F_MEDIATED_REGULATION_OF_DNA_REPLICATION                    4.623461e-08
## REACTOME_DEPOSITION_OF_NEW_CENPA_CONTAINING_NUCLEOSOMES_AT_THE_CENTROMERE 2.053522e-07
## REACTOME_G2_M_CHECKPOINTS                                              2.093024e-07
## REACTOME_G1_S_SPECIFIC_TRANSCRIPTION                                   7.194960e-07
## REACTOME_CELL_CYCLE                                                    1.933004e-06
##                                                                               FDR
## REACTOME_MITOTIC_PROMETAPHASE                                          3.603532e-06
## REACTOME_E2F_MEDIATED_REGULATION_OF_DNA_REPLICATION                    1.528054e-05
## REACTOME_DEPOSITION_OF_NEW_CENPA_CONTAINING_NUCLEOSOMES_AT_THE_CENTROMERE 3.458721e-05
## REACTOME_G2_M_CHECKPOINTS                                              3.458721e-05
## REACTOME_G1_S_SPECIFIC_TRANSCRIPTION                                   9.511738e-05
## REACTOME_CELL_CYCLE                                                    2.129526e-04
```

As reported by McDavid et al., a substantial number of the significant terms relate to cell cycle. We next have a more detailed look at the inter-gene correlation.

```
nTop = 10
soCam = sort(res[,5], index.return=T)
sets_filter = sets_indices[rownames(res)[soCam$ix[1:nTop]]]
corr = unlist(lapply(sets_filter, function(x)mean(cor(T_cell_matrix_corrected[TcellCdat$clusterid==0,x]
corrI = unlist(lapply(sets_filter, function(x)(interGeneCorrelation(y=t(T_cell_matrix_corrected[,x]), d
```

```
ggplot()+geom_boxplot(mapping=aes(x='CAMERA/McDavid et al', y=corrI))+ylab("Inter-gene correlation")+xla
```

If we look at the inter-gene correlation of the top 10 modules (after accounting for the clusters), we find - as expected - that it is very low, with a median of 0.01. Next, we compare this to the correlation within the modules we identified.

First retrieve the gene sets from topGO.

```
#load DE genes
DEgenes = read.xls('~/Dropbox/SC_RNAseq_Cell_Cycle/Final Version/submission/supplementary files/Supplem

all_genes <- as.factor(as.integer(sym_names[is_expressed] %in% DEgenes))
names(all_genes) <- sym_names[is_expressed]

#retrieve annotation data froom GO
tgd <- new( "topGOdata", ontology="BP", allGenes = as.factor(all_genes),nodeSize=5,
```

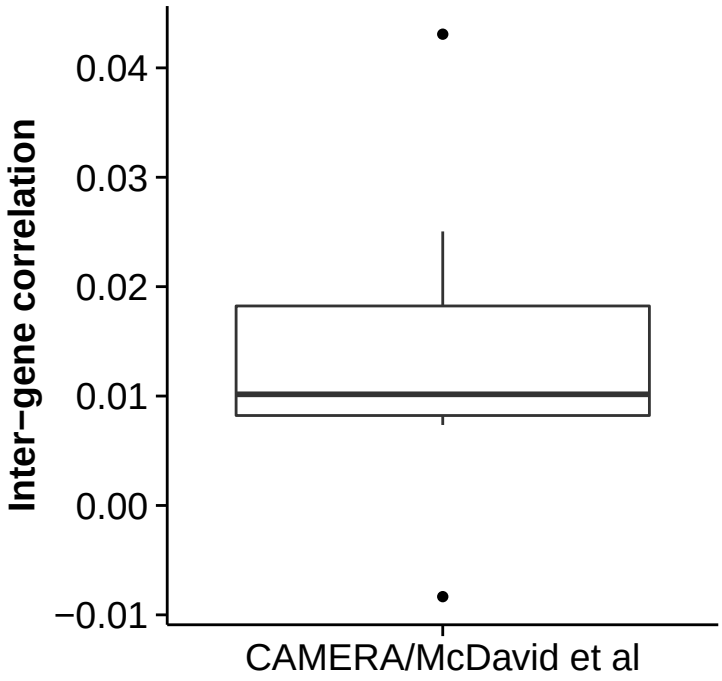


Figure 2: Inter-gene Correlation CAMERA/McDavid et al.

```
            annot=annFUN.org, mapping="org.Mm.eg.db",ID='symbol')
go <- usedGO(tgd)
ann.genes = genesInTerm(tgd, go)
#names(ann.genes) = Term(names(ann.genes))

sets_indicesBNC <- ids2indices(ann.genes, (colnames(T_cell_matrix_corrected)))
#names(sets_indicesBNC) = Term(names(sets_indicesBNC))

resultTopGO <- runTest(tgd, algorithm = "elim", statistic = "Fisher" )
tab = GenTable( tgd, resultTopGO, topNodes=nTop )
```

Now we can compare the intergene correlation in the topGO modules.

```
sets_filterBNC = sets_indicesBNC[tab[,1]]
corrBNC = unlist(lapply(sets_filterBNC, function(x)mean(cor(T_cell_matrix_corrected[TcellCdat$clusterid=
corrIBNC = unlist(lapply(sets_filterBNC, function(x)(interGeneCorrelation(y=t(T_cell_matrix_corrected[,:

dfCorr = data.frame(corr = c(corr, corrBNC),corrI = c(corrI, corrIBNC), label = c(rep("CAMERA",nTop), re
```

```
ggplot(data=dfCorr) + geom_boxplot(mapping  = aes(y=corrI, x=label))+xlab("Gene sets")+ylab("Inter-gene
```

With a median of 0.101, the inter-gene correlation for the gene set identified by topGO was about 10 times higher than in those sets identified by CAMERA (again after controlling for the clusters). We also explore how many of the genes that were differntially expressed between the cluster, are present in the modules identified by topGO and CAMERA respectively. As CAMERA penalizes inter-gene correlations, also weak signatures such as remaining cell-cycle signals can appear to be enriched. We investigated an interaction effect between cell cycle and differntiation that could potentially explain this effect in our primary publication . Therefore, we also explore the number of these interaction genes present in the various sets.

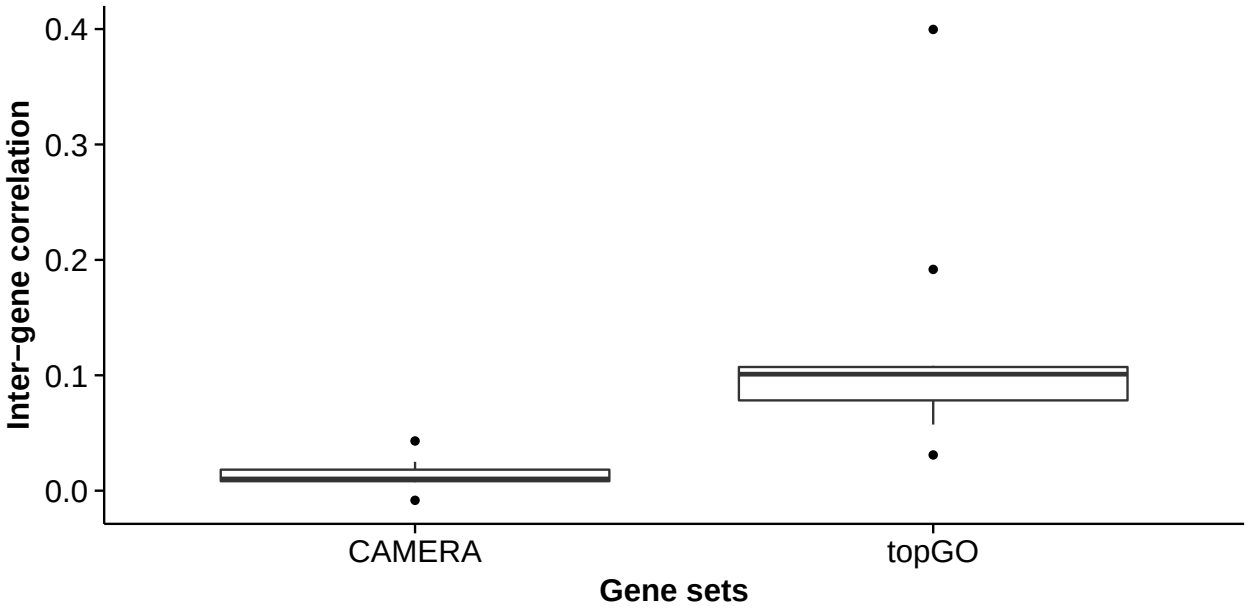


Figure 3: Inter-gene correlation for CAMERA/McDavid et al. and topGP/Buettner et al.

```
int_genes <- read.xls('~/Dropbox/SC_RNAseq_Cell_Cycle/Final Version/submission/supplementary files/Suppl


numTop = c()
numTerm = c()
numTop_noInt = c()
DE_in_Modules = c()
in_Modules = c()

topRank = length(DEgenes)+1
for(j in 1:nTop){


idx_term = j

idx_set = match(geneID[unlist(sets_indices[resCAMERASorted$set[idx_term]])],geneID)
idx_setDE = match(geneID[unlist(sets_indices[resCAMERASorted$set[idx_term]])],DEgenes)



idx_inter_set = idx_set[na.omit(match(as.character(int_genes),geneID[idx_set]))]

num_highInter = length(intersect(as.character(int_genes),DEgenes[idx_setDE]))

numTop[j] = length(na.omit(idx_setDE))
numTerm[j] = length(na.omit(idx_set))
numTop_noInt[j] = numTop[j]-num_highInter#sum(idx_inter_set<topRank)


DE_in_Modules = c(DE_in_Modules,intersect(DEgenes, geneID[unlist(sets_indices[resCAMERASorted$set[idx_t
```

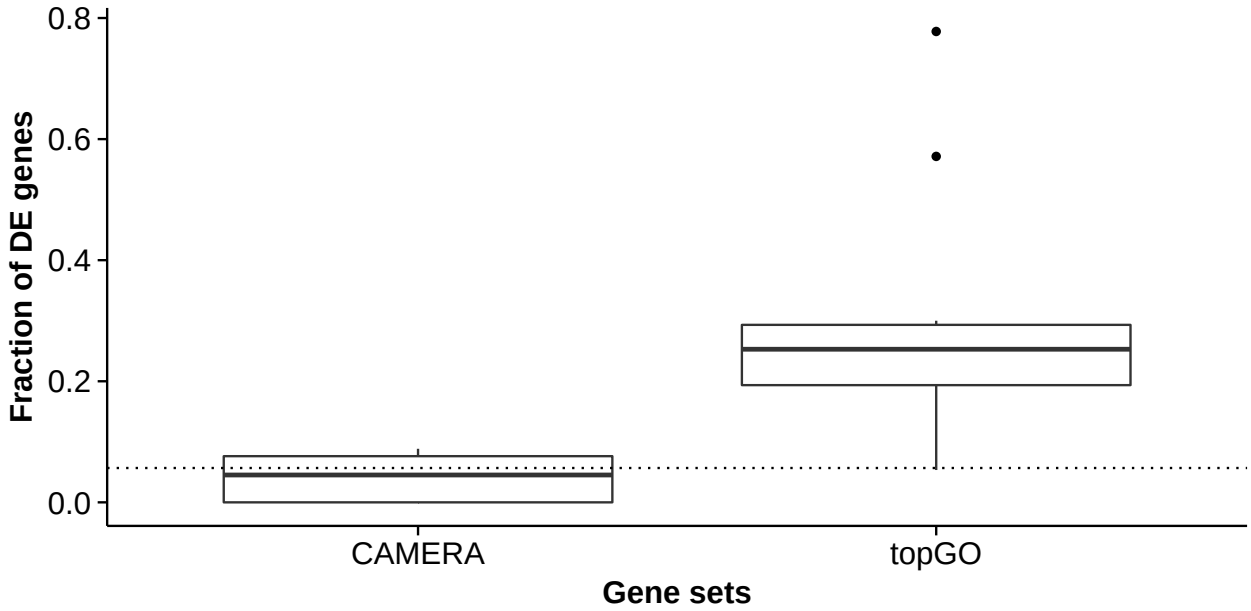
```
}
nDEinModules = length(unique(DE_in_Modules))
df = data.frame(frac = c((numTop/numTerm)[1:nTop], (tab[,4]/tab[,3])[1:nTop]), label = c(rep("CAMERA",n

resBNC = cbind(numTerm,numTop,numTop_noInt)
resBNC
```

```
##       numTerm numTop numTop_noInt
##  [1,]      66      4            3
##  [2,]      27      0            0
##  [3,]      19      1            0
##  [4,]      36      0            0
##  [5,]      14      0            0
##  [6,]     257     21            7
##  [7,]      53      2            1
##  [8,]     226     20            7
##  [9,]     135     11            4
## [10,]      13      0            0
```

This illustrates that only a small number of DE genes are annotated to the gene sets identified by McDavid et al. More specifically, in the top 10 modules only 21 genes overlapped the annotated genes. When removing the interaction genes, this number decreased to 7 genes.

```
ggplot(data=df) + geom_boxplot(mapping  = aes(y=frac, x=label))+xlab("Gene sets") +ylab("Fraction of DE
  geom_hline(aes(yintercept=rep(length(DEgenes)/length(geneID), 2*nTop)) ,linetype="dotted")
```



The dotted lines illustrates the number of DE genes expected by chance.

Remark on T-cell clustering

McDavid et al. report they were unable to replicate the clustering for the corrected data using dimensionality reduction methods other than GPLVM. While GPLVM perfroms best on these data, we can also find a largely similar clustering using diffusion maps.

```
diffmap = DiffusionMap(T_cell_matrix_corrected)

dDiffMap <- qplot(diffmap@eigenvectors[,1], diffmap@eigenvectors[,2], colour=TcellCdat$clusterid) + xlab
```

```
dDiffMap
```

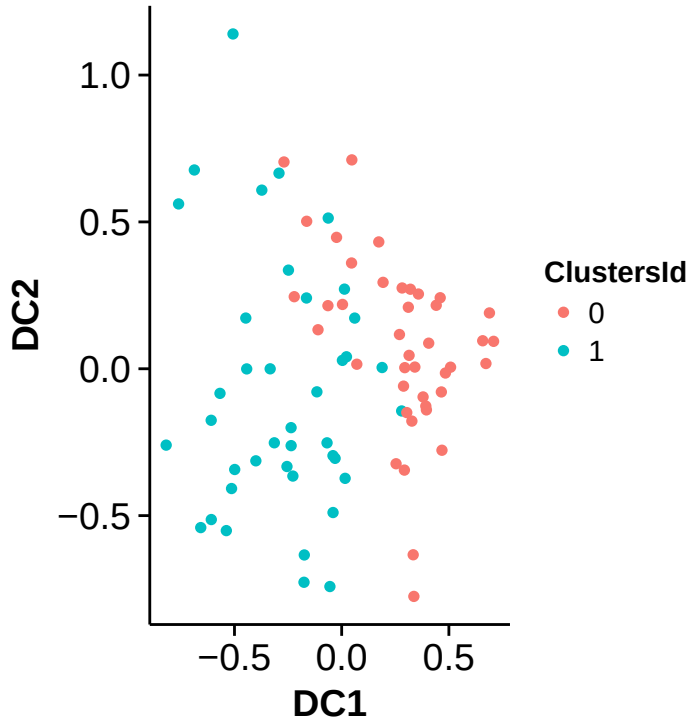


Figure 4: Diffusion Map

Session Info

This was generated using

```
sessionInfo()
```

```
## R version 3.1.3 (2015-03-09)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.5 (Yosemite)
##
## locale:
## [1] de_DE.UTF-8/de_DE.UTF-8/de_DE.UTF-8/C/de_DE.UTF-8/de_DE.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] DESeq_1.18.0         lattice_0.20-31      locfit_1.5-9.1
```

```
##  [4] scLVM_0.99.2         rPython_0.0-5        RJSONIO_1.3-0
##  [7] topGO_2.18.0         SparseM_1.7          GO.db_3.0.0
## [10] cowplot_0.4.0        destiny_0.9.1        BASiCS_0.2.0
## [13] GSEABase_1.28.0      graph_1.44.1         annotate_1.44.0
## [16] XML_3.98-1.1         limma_3.22.7         org.Mm.eg.db_3.0.0
## [19] RSQLite_1.0.0        DBI_0.3.1            AnnotationDbi_1.28.2
## [22] GenomeInfoDb_1.2.5   IRanges_2.0.1        S4Vectors_0.4.0
## [25] Biobase_2.26.0       BiocGenerics_0.12.1  gplots_2.17.0
## [28] gdata_2.17.0         ggplot2_1.0.1        data.table_1.9.4
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6         car_2.0-25           caTools_1.17.1
##  [4] chron_2.3-47         class_7.3-12         coda_0.17-1
##  [7] codetools_0.2-11     colorspace_1.2-6     DEoptimR_1.0-2
## [10] digest_0.6.8         e1071_1.6-4          evaluate_0.7
## [13] FNN_1.1              formatR_1.2          genefilter_1.48.1
## [16] geneplotter_1.44.0   grid_3.1.3           gtable_0.1.2
## [19] gtools_3.4.2         htmltools_0.2.6      igraph_0.7.1
## [22] KernSmooth_2.23-14   knitr_1.10.5         labeling_0.3
## [25] lme4_1.1-9           magrittr_1.5         MASS_7.3-40
## [28] Matrix_1.2-0         MatrixModels_0.4-1   mgcv_1.8-6
## [31] minqa_1.2.4          munsell_0.4.2        nlme_3.1-120
## [34] nloptr_1.0.4         nnet_7.3-9           pbkrtest_0.4-2
## [37] plyr_1.8.3           proto_0.3-10         proxy_0.4-14
## [40] quantreg_5.19        RColorBrewer_1.1-2   Rcpp_0.12.1
## [43] reshape2_1.4.1       rmarkdown_0.5.1      robustbase_0.92-4
## [46] scales_0.2.5         scatterplot3d_0.3-35 sp_1.1-0
## [49] splines_3.1.3        statmod_1.4.21       stringi_0.5-5
## [52] stringr_1.0.0        survival_2.38-1      tools_3.1.3
## [55] vcd_1.3-2            VIM_4.1.0            xtable_1.7-4
## [58] yaml_2.1.13
```