Chapter 7

Differential Pathway Analysis

Jean Fan

Abstract

Integrating prior knowledge of pathway-level information can enhance power and facilitate interpretation of gene expression data analyses. Here, we provide a practical demonstration of the value of gene set or pathway enrichment testing and extend such techniques to identify and characterize transcriptional subpopulations from single-cell RNA-sequencing data using pathway and gene set overdispersion analysis (PAGODA).

Key words Single cell, Pathway, Gene set enrichment analysis, Differential expression analysis, Clustering

1 Introduction

Identifying genes that exhibit significant differences among two or more biological states, conditions, or cell-types is integral to understanding the putative molecular bases for phenotypic variation. Determining whether individual genes exhibit significant expression differences between conditions can be achieved using differential gene expression analysis [1]. However, when gene expression data are noisy and biological signals are weak, testing individual genes for differences may not provide any statistically significant results. In particular for single-cell RNA-seq data, such a differential expression analysis is often complicated by high levels of technical noise and intrinsic biological stochasticity in the data. As such, application of previous differential expression analysis approaches developed for bulk RNA-seq may not always be suitable [2]. While methods for differential expression analysis specifically tailored to single-cell RNA-seq data have been developed [3, 4], alternatively grouping genes into biologically-relevant modules such as pathways may greatly enhance statistical power and improve our ability to identify true biological signal [5, 6]. In this chapter, we will discuss how to take a pathway-informed approach to differential

Guo-Cheng Yuan (ed.), *Computational Methods for Single-Cell Data Analysis*, Methods in Molecular Biology, vol. 1935, https://doi.org/10.1007/978-1-4939-9057-3_7,

expression analysis and apply it to single-cell RNA-seq data to identify and characterize transcriptional subpopulations.

Gene set or pathway enrichment analysis is a computational approach that determines whether an a priori defined set of genes such as a pathway shows statistically significant, concordant differences between two biological states. Gene set or pathway enrichment analysis is particularly powerful when genes individually do not exhibit a statistically significant difference between two biological states, but, when grouped together, show statistically significant concordant differences. For example, when performing differential expression analysis, one common approach is to use a significance cutoff to identify a limited number of the most interesting genes for further research and interpretation. Gene set or pathway enrichment analysis takes an alternative approach by focusing on cumulative expression changes of multiple genes as a group, thus shifting the focus from individual genes to groups of genes. By looking at several genes at once, such an approach can identify gene sets or pathways that have several genes each change a small amount, but in a coordinated way, which may reach statistical significance even when individual gene expression changes are quite small and insufficiently significant.

There are many different methods to perform gene set or pathway enrichment analysis, from hypergeometric distribution tests [7, 8] to permutation-based approaches [5]. One popular method, aptly named Gene Set Enrichment Analysis (GSEA) [5], tests for enrichment using a permutation-based approach. In GSEA, first, genes are ranked such as based on a measure of each gene's differential expression with respect to the two conditions. Then the entire ranked list is used to assess how the genes of each gene set are distributed across the ranked list by walking down the ranked list of genes, increasing a running-sum statistic when a gene belongs to the set, and decreasing it when the gene does not (Fig. 1). The enrichment score is the maximum deviation from zero encountered during the walk. The score reflects the degree to which the genes in a gene set are overrepresented at the top or bottom of the entire ranked list of genes. A set that is not enriched will have its genes spread more or less uniformly through the ranked list. An enriched set, on the other hand, will have a larger portion of its genes at one or the other end of the ranked list. The extent of enrichment is captured mathematically as the score statistic. The statistical significance of the score can then be estimated using permutation, whereby enrichment scores are computed for random gene sets of the same size as the tested gene set. This randomization is repeated many times to produce an empirical null distribution of scores. The nominal p-value estimates the statistical significance of a single gene set's score based on the permutation-generated null distribution.

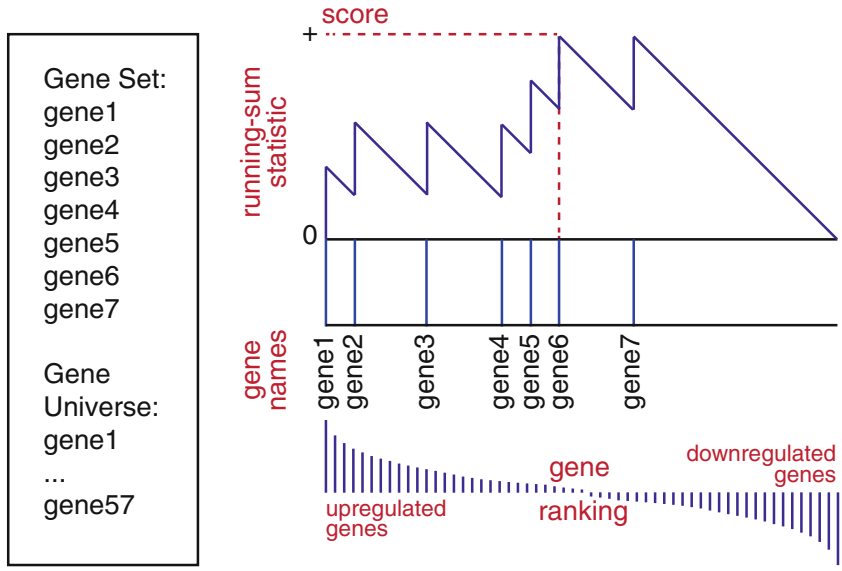

Fig. 1 A standard gene set enrichment plot. Genes in the gene universe are ranked according to a differential expression statistic from most upregulated to most downregulated. A running-sum statistic then traverses the ranked list and increments the enrichment score statistic upon reaching a gene within the gene set of interest

2 Materials

All programming will be done using the R statistical programming language [9].

2.1 *Liger R Package*

In Subheading 3.1, we will perform gene set enrichment analysis using the Lightweight Iterative Gene set Enrichment in R (liger) package, an R implementation of the GSEA algorithm [5]. liger can be installed from CRAN using the following command in R:

```
install.packages("liger")
```

2.2 *Scde R Package*

In Subheadings 3.2 and 3.3, we will perform pathway and gene set overdispersion analysis (PAGODA) using the Single Cell Differential Expression (scde) package. Scde can be installed from Bioconductor using the following command in R:

```
# try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite("scde")
```

3 Methods

3.1 Enhancing Statistical Power by Incorporating Pathway-Level Information

To demonstrate the utility of gene set or pathway enrichment analysis, we will use a simulated dataset. Specifically, we will simulate a weak differential expression within a known gene set between two biological samples. We will show that while differential expression analysis is not able to pick up these genes as significantly differentially expressed, a gene set enrichment analysis will be able to pick up significant enrichment.

1. First, we will load the liger package.

```
library(liger)
```

2. Load a gene set based on Gene Ontology (GO) terms.

```
# load gene set
data("org.Hs.GO2Symbol.list")
```

We can look into the newly loaded `org.Hs.GO2Symbol.list` object. Notice that it is a list of GO ids for various gene sets. Each list contains the human HUGO symbols of genes within that gene set. Note, in this manner, alternative gene sets such as MSigDB [10] or KEGG or even custom gene sets can also be created and used.

```
head(org.Hs.GO2Symbol.list)
## $`GO:0000002`
##  [1] "AKT3"     "C10orf2"  "DNA2"     "LIG3"     "MEF2A"    "MGME1"
##  [7] "MPV17"    "OPA1"     "PID1"     "PRIMPOL"  "SLC25A33" "SLC25A36"
## [13] "SLC25A4"  "STOML2"   "TYMP"
...
```

3. To simulate a weak differential expression within a known gene set between two biological samples, we will first simulate random gene expression for 100 cells. We will create a matrix containing all genes and simulate gene expression by drawing from a normal distribution with mean = 0, and sd = 3. Alternatively, your own normalized single-cell expression data can be substituted in at this step.

```
# set random seed to ensure reproducibility
set.seed(0)
# get universe of genes
universe <- unique(unlist(org.Hs.GO2Symbol.list))
# make random data
Nsamples <- 100 # 100 cells
Mgenes <- length(universe)
mat <- matrix(rnorm(Mgenes*Nsamples, mean=0, sd=3), Mgenes, Nsamples)
rownames(mat) <- universe
```

Next, we will first pick a gene set.

```
# get genes in gene set GO:0000002
gs <- org.Hs.GO2Symbol.list[["GO:0000002"]]
# genes
print(gs)
##  [1] "AKT3"     "C10orf2"  "DNA2"     "LIG3"     "MEF2A"    "MGME1"
##  [7] "MPV17"    "OPA1"     "PID1"     "PRIMPOL"  "SLC25A33" "SLC25A36"
## [13] "SLC25A4"  "STOML2"   "TYMP"
```

We will split our 100 cells into 2 groups to represent two biologically different states.

```
# two biological states (groups)
group <- factor(c(rep(1, Nsamples/2), rep(2, Nsamples/2)))
names(group) <- colnames(mat) <- paste0('sample', 1:Nsamples)
```

Now, we will simulate upregulation of genes from our selected gene set in cells belonging to group 1. Genes from our selected gene set in cells belonging to group 1, rather than being drawn from a normal distribution with mean = 0 and sd = 3, will instead have an increased mean = 2.25, and sd = 5. We will also remove negative values in our simulation to keep simulated expression values interpretable.

```
# simulate upregulation of gene set in group 1
mat[gs, group==1] <- rnorm(length(gs)*sum(group==1), mean=2.25, sd=5)
# make more realistic; can't have negative gene expression
mat[mat < 0] <- 0
```

4. Now we can visualize the expression of our simulated upregulated genes along with 50 other non-upregulated genes using a heatmap (Fig. 2). We will also color the column side bar of our heatmap using the cell group labels, with group 1 cells labeled in red, and group 2 cells labeled in blue. Similarly, we will color the row side bar in green if the gene is within our selected gene set and black if not.

```
# we can visualize this weak differential expression in a heatmap
# visualize weakly differentially expressed genes and another 50 genes
vi <- c(gs, universe[1:50])
# label supposedly differentially expressed genes
heatmap(mat[vi,], Rowv=NA, Colv=NA, scale="none",
        col=colorRampPalette(c("blue", "white", "red"))(100),
        RowSideColors = c('black', 'green')[as.factor(vi %in% gs)],
        ColSideColors = c('red', 'blue')[group])
```

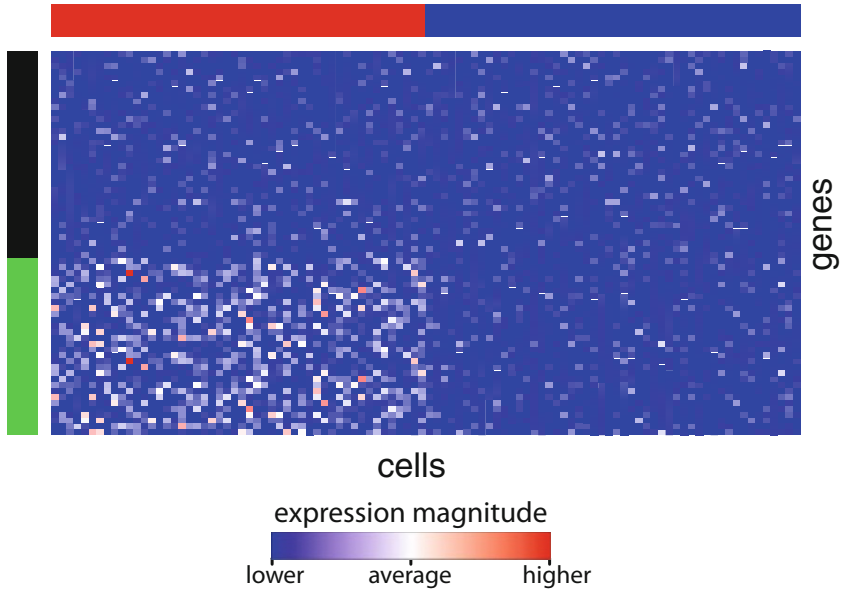

Fig. 2 Gene expression heatmap for select simulated genes. Rows are genes and columns are cells. Gene expression is colored using a color ramp from blue to white to red, with highly expressed genes colored in red and lowly expressed genes in blue. Column side bar is colored using the cell group labels, with group 1 cells labeled in red, and group 2 cells labeled in blue. Row side bar is colored in green if the gene is within our selected gene set and black if not

Although we simulated the green row side color annotated genes to be upregulated in the red column side color annotated samples compared to the blue column side color annotated samples, even visually, it is somewhat difficult to tell which genes are differentially expressed.

5. We can also quantify the extent of the differential expression between our two biological states using a t-test.

```
# run differential expression analysis using simple t-test
vals.info <- lapply(1:nrow(mat), function(i) {
    pv <- t.test(
        mat[i, group==1],
        mat[i, group==2]
    )
    return(list(val=pv$statistic, p=pv$p.value))
})
vals <- unlist(lapply(vals.info, function(x) x$val))
p <- unlist(lapply(vals.info, function(x) x$p))
names(p) <- names(vals) <- rownames(mat)
```

6. Because we are testing many genes, we have to apply multiple-testing correction. We will use a Bonferroni correction [11].

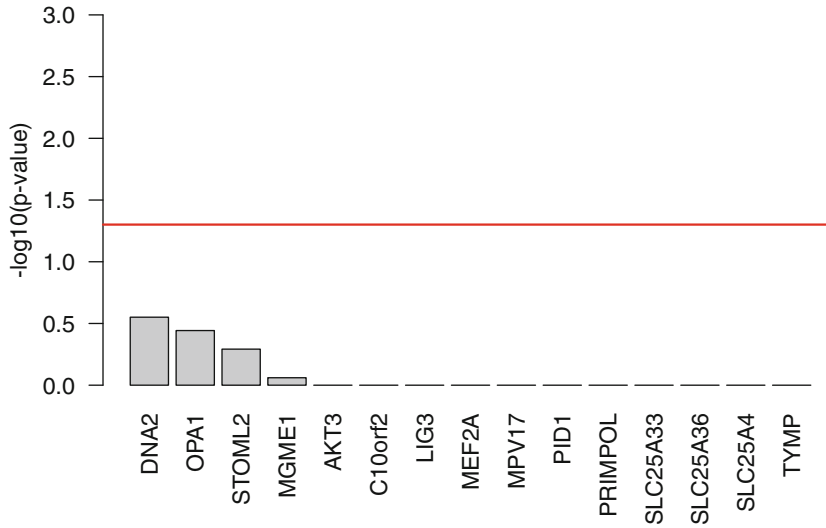

Fig. 3 Differential expression analysis results for select simulated genes. Barplot shows -log10(*p*-value) for each gene. Red line shows the $p = 0.05$ significance threshold. Note none of the tested genes passes the significance threshold

```
p.adj <- p.adjust(p, method="bonferroni") # multiple-testing correction
names(p.adj) <- rownames(mat)
```

7. We can now visualize the final -log10(*p*-values) using a barplot (Fig. 3). We will use a red line to indicate the common $p < 0.05$ significance threshold. Significant genes should have bars that pass the red line.

```
barplot(sort(-log10(p.adj[gs]), decreasing=TRUE), ylim=c(0, 3), las=2)
abline(h = -log10(0.05), col="red")
```

Unfortunately, none of the genes, including those we simulated to be differentially expressed, were actually picked up as significantly differentially expressed after multiple-testing correction (with corrected *p*-values <0.05). In a real-world situation, we may be tempted to end our analysis here and conclude that since nothing is significantly differentially expressed between the two biological states there is no significant difference.

However, we can still perform gene set or pathway enrichment analysis on a priori defined gene sets to look for statistically significant concordant differences.

8. We will perform such analyses using liger for 10 Gene Ontology gene sets in org.Hs.GO2Symbol.list, including GO:0000002.

```
# run iterative bulk gsea on our true gene set and 9 other gene sets as
test
gseaVals <- iterative.bulk.gsea(
    values = vals,
    set.list = org.Hs.GO2Symbol.list[1:10],
    rank=TRUE)
## initial: [1e+02 - 3] [1e+03 - 1] [1e+04 - 1] done
print(gseaVals)
##                 p.val      q.val     sscore       edge
## GO:0000002 0.00009999 0.00059994  2.5584741  2.0848842
## GO:0000003 0.66336634 0.66336634  0.4924230  0.3374948
## GO:0000012 0.11888112 0.25774226 -0.9737758 -0.1256842
## GO:0000014 0.24752475 0.36831683  0.6518057 -0.5915193
## GO:0000018 0.30693069 0.36831683 -0.7279604  0.9366813
## GO:0000022 0.12887113 0.25774226  0.9455950 -0.4223886
```

9. We can then identify significantly enriched gene sets as those with a q-value <0.05.

```
# identify significantly enriched gene sets
gseaSig <- rownames(gseaVals[gseaVals$q.val < 0.05,])
print(gseaSig)
## [1] "GO:0000002"
```

Indeed, we recover GO:0000002 as a significantly enriched gene set!

10. We can visualize a standard gene set enrichment plot for this gene set (Fig. 4).

```
# look at plots
for(i in seq_along(gseaSig)) {
  gs <- org.Hs.GO2Symbol.list[[gseaSig[i]]]
  gsea(values=vals, geneset=gs, mc.cores=1, plot=TRUE, rank=TRUE)
}
```

So, although no individual gene was found to be statistically significantly differentially expressed between our two biological states, gene set and pathway enrichment analysis identified a significantly enriched gene set, GO:0000002, which is exactly the gene set that we simulated to show concordant differences. By looking for coordinated changes in genes within these a priori defined gene sets, we are able to increase our statistical power to identify differences between our two biological states.

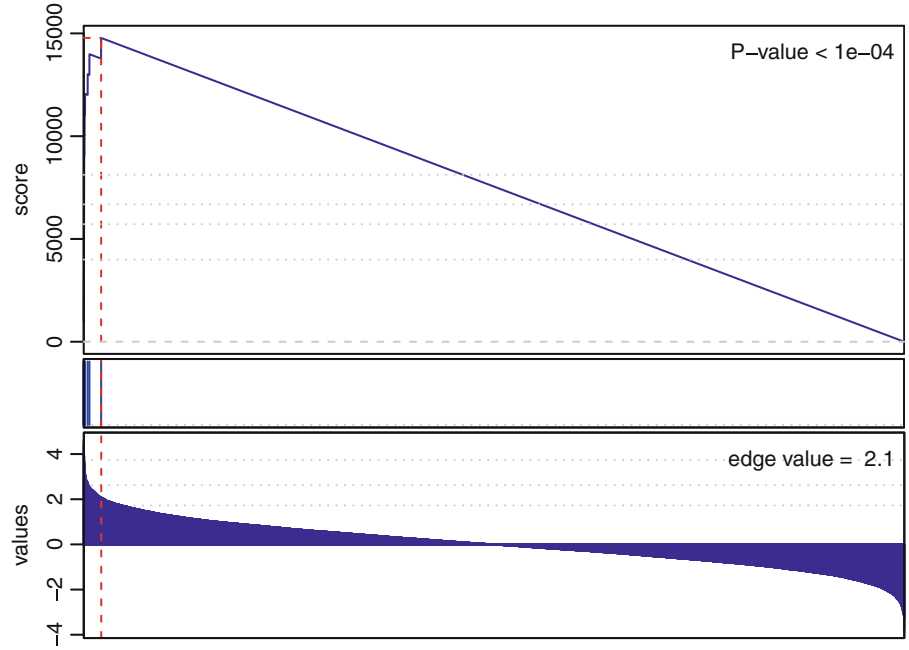

Fig. 4 Gene set enrichment plot for gene set GO:0000002 demonstrates significant enrichment as simulated

3.2 Applying a Pathway-Integrated Approach with Pathway and Gene Set Overdisperrsion Analysis

Gene set testing with methods such as liger can be used for differential expression analysis to increase statistical power and uncover likely functional interpretations. However, such testing requires knowledge of biological conditions or subpopulations for comparison. To identify these transcriptionally distinct subpopulations, a similar rationale can be applied in single-cell RNA-seq data analysis. Highly variable genes may partition cells into transcriptionally distinct subpopulations but carry consideration uncertainty as observed variability in gene expression may be the result of technical artifacts such as drop-outs. Yet whereas variability in the expression of a single gene may be noisy, coordinated upregulation of many genes within a gene set or pathway in the same subset of cells could provide a prominent signature to distinguish subpopulations.

Pathway And Gene set Over-Dispersion Analysis (PAGODA) [6] looks for coordinated expression variability of genes in both annotated pathways and automatically detected "de novo" gene sets. PAGODA then uses this gene set and pathway-level information to cluster cells into transcriptional subpopulations.

Briefly, PAGODA first estimates the effective sequencing depth, drop-out rate, and amplification noise of each cell using a previously described mixture-model approach with minor enhancements. Using these models, the observed expression variance of each gene is renormalized on the basis of the expected genome-

wide variance at the appropriate expression magnitude. PAGODA then examines an extensive panel of gene sets to identify those showing a statistically significant excess of coordinated variability. Gene sets can include annotated pathways, such as Gene Ontology (GO) categories, as well as clusters of transcriptionally correlated genes found in a given data set ("de novo" gene sets). The prevalent transcriptional signature of each gene set is captured by its first principal component (PC), with weighted PCA used to adjust for technical noise. If the amount of variance explained by the first PC of a given gene set is significantly higher than expected, the gene set is considered to be "overdispersed. "PCs from the resulting significantly overdispersed gene sets are combined to form a single "aspect" of heterogeneity to provide a nonredundant view of transcriptional heterogeneity to users through an interactive web browser interface.

To demonstrate the utility of PAGODA, we will continue our exploration of our simulated dataset. Note, to run PAGODA using your own single-cell RNA-seq data, see Subheading 3.3 for step-by-step instructions on how to go from gene expression counts to the appropriate variance-normalized gene expression matrix inputted into the pathway overdispersion testing step.

1. We will first show how unbiased hierarchical clustering on our simulated raw data fails to cluster our true groups together (Fig. 5).

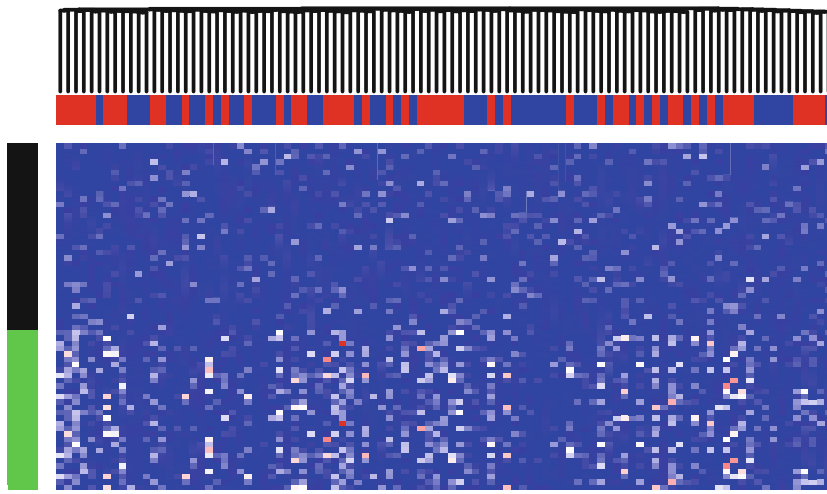

Fig. 5 Gene expression heatmap with cells grouped by hierarchical clustering shows inconsistency with cell group labels. Rows are genes and columns are cells. Gene expression is colored using a color ramp from blue to white to red, with highly expressed genes colored in red and lowly expressed genes in blue. Column side bar is colored using the cell group labels, with group 1 cells labeled in red, and group 2 cells labeled in blue. Row side bar is colored in green if the gene is within our selected gene set and black if not

```
# just cluster by all genes
hc <- hclust(dist(t(mat)))
heatmap(mat[vi,], Rowv=NA, Colv=as.dendrogram(hc), scale="none",
        col=colorRampPalette(c("blue", "white", "red"))(100),
        RowSideColors = c('black', 'green')[as.factor(vi %in% gs)],
        ColSideColors = c('red', 'blue')[group])
```

The cells are ordered by unbiased hierarchical clustering and we do not see any segregation of our two cell group labels. However, we can integrate pathway-level information to enhance our signal and enable proper separation of our two simulated cell groups.

2. To run PAGODA, we will load the scde package and format our simulated data into the appropriate format. Note, to run PAGODA using your own single-cell RNA-seq data, additional functions are available for error-modeling and normalization from gene expression counts (*See*.

```
library(scde)

# format data to pipe into PAGODA
varinfo <- list()
varinfo$mat <- mat
matw <- matrix(1, nrow(mat), ncol(mat)) # equal weighting
rownames(matw) <- rownames(mat)
colnames(matw) <- colnames(mat)
varinfo$matw <- matw
```

3. We will compute PCs for a set of 10 pathways and cluster based on such pathway-level expression.

```
go.env <- list2env(org.Hs.GO2Symbol.list[1:10]) # just use first 10
pathways
# test pathways for overdispersion
pwpca <- pagoda.pathway.wPCA(varinfo, go.env, n.components = 1, n.cores =
1)
df <- pagoda.top.aspects(pwpca, return.table = TRUE, plot = FALSE,
z.score = 1.96)
head(df)
##         name npc   n     score         z     adj.z sh.z adj.sh.z
## 1 GO:0000002   1  15 212.82838 152.43638 152.42917   NA       NA
## 3 GO:0000018   1  34  31.30615  50.76668  50.75870   NA       NA
## 2 GO:0000003   1 300  10.29349  50.41152  50.41152   NA       NA

tam <- pagoda.top.aspects(pwpca, z.score = qnorm(0.01/2, lower.tail =
FALSE))
```

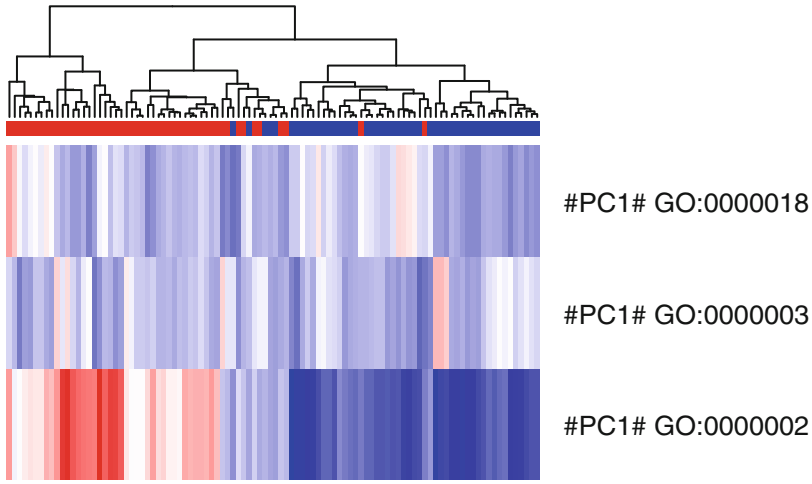

Fig. 6 Pathway expression heatmap with cells grouped by hierarchical clustering shows consistency with cell group labels. Rows are pathways and columns are cells. Pathway expression, summarized by the first principal component (PC1) of gene expressions for genes within the pathway, is colored using a color ramp from blue to white to red. Column side bar is colored using the cell group labels, with group 1 cells labeled in red, and group 2 cells labeled in blue

4. Now, we can cluster our cells based on their pathway-level expression patterns using hierarchical clustering and visualize the results as a heatmap (Fig. 6).

```
# unbiased clustering on pathway information
hc2 <- hclust(dist(t(tam$xv)))
heatmap(tam$xv, Rowv=NA, Colv=as.dendrogram(hc2), scale="none",
        col=colorRampPalette(c("blue", "white", "red"))(100),
        ColSideColors = c('red', 'blue')[group], mar=c(5,15))
```

And indeed, we can see that the pathway-integrated clustering better separates our two simulated groups.

3.3 Pathway and Gene Set Overdisperrsion Analysis with Single-Cell RNA-Seq Data

For a more realistic demonstration, we will analyze single-cell RNA-seq data from Pollen et al. [12]. The error models PAGODA uses are based off of count-based processes and therefore the inputted data will be a matrix of read counts.

1. We can load the read count table and cell group annotations using data("pollen") call. The columns are cells and the rows are genes. Some additional filters are also applied to remove poor cells and non-detected genes. Your own single-cell RNA--seq data can be substituted at this step as well.

```
library(scde)
data(pollen)
# remove poor cells and genes
cd <- clean.counts(pollen)
# check the final dimensions of the read count matrix
dim(cd)
## [1] 11310    64
```

For visualizations later, we will translate group and sample source data from the original publication [12] into color codes.

```
x <- gsub("^Hi_(.*)_.*", "\\1", colnames(cd))
l2cols <- c("coral4", "olivedrab3", "skyblue2",
"slateblue3")[as.integer(factor(x, levels = c("NPC", "GW16", "GW21",
"GW21+3")))]
```

2. Next, we'll construct error models for individual cells. Here, we use a k-nearest neighbor model fitting procedure implemented by knn.error.models() method. This is a relatively noisy dataset (non-UMI), so we raise the min.count.threshold to 2 (minimum number of reads for the gene to be initially classified as a non-failed measurement), requiring at least 5 non-failed measurements per gene. We're providing a rough guess to the complexity of the population, by fitting the error models based on 1/4 of most similar cells (i.e., guessing there might be ~4 subpopulations). Note, this step takes a considerable amount of time unless multiple cores are used. We highly recommend use of multiple cores. You can check the number of available cores available using detectCores().

```
knn <- knn.error.models(cd, k = ncol(cd)/4, n.cores = 1,
min.count.threshold = 2, min.nonfailed = 5, max.model.plots = 10)
```

3. In order to accurately quantify excess variance or overdispersion, we must normalize out expected levels of technical and intrinsic biological noise. Briefly, variance of the NB/Poisson mixture processes derived from the error modeling step is modeled as a chi-squared distribution using adjusted degrees of freedom and observation weights based on the drop-out probability of a given gene. We will normalize variance, trimming 3 most extreme cells and limiting maximum adjusted variance to 5.

```
varinfo <- pagoda.varnorm(knn, counts = cd, trim = 3/ncol(cd),
max.adj.var = 5, n.cores = 1, plot = TRUE)
```

4. Even with all the corrections, sequencing depth or gene coverage is typically still a major aspect of variability. In most studies, we would want to control for that as a technical artifact (exceptions are cell mixtures where subtypes significantly differ in the amount of total mRNA). We will control for the gene coverage (estimated as a number of genes with nonzero magnitude per cell) by normalizing out that aspect of cell heterogeneity:

```
varinfo <- pagoda.subtract.aspect(varinfo, colSums(cd[,
rownames(knn)]>0))
```

5. As mentioned previously, in order to detect significant aspects of heterogeneity across the population of single cells, PAGODA identifies pathways and gene sets that exhibit statistically significant excess of coordinated variability. Specifically, for each gene set, we will test whether the amount of variance explained by the first principal component significantly exceeds the background expectation. We can test both predefined gene sets as well as "de novo" gene sets whose expression profiles are well correlated within the given dataset.

   For predefined gene sets, we'll use the GO annotations we previously loaded from liger.

```
# in case you didn't load it previously, load it now
library(liger)
data("org.Hs.GO2Symbol.list")
go.env <- org.Hs.GO2Symbol.list
# remove GOs with too few or too many genes
go.env <- clean.gos(go.env)
# convert to an environment
go.env <- list2env(go.env)
```

   Now, we can calculate weighted first principal component magnitudes for each GO gene set in the provided environment and evaluate the statistical significance of their overdispersion.

```
pwpca <- pagoda.pathway.wPCA(varinfo, go.env, n.components = 1, n.cores =
1)
df <- pagoda.top.aspects(pwpca, return.table = TRUE, plot = FALSE,
z.score = 1.96)
head(df)
##            name npc  n    score         z     adj.z
## 339  GO:0003179   1 10 3.495767 11.108780 10.760218
## 338  GO:0003170   1 10 3.495767 11.108780 10.760218
## 3570 GO:0060563   1 12 3.220725 10.643172 10.297292
## 1829 GO:0030426   1 39 3.134488 14.644926 14.338584
## 1302 GO:0014009   1 10 3.105600  9.656705  9.307366
## 1830 GO:0030427   1 40 3.093050 14.530866 14.223476
```

- The z column gives the Z-score of pathway over-dispersion relative to the genome-wide model (Z-score of 1.96 corresponds to *P*-value of 5%, etc.).
- "z.adj" column shows the Z-score adjusted for multiple hypothesis (using Benjamini-Hochberg correction).
- "score" gives observed/expected variance ratio.
- "sh.z" and "adj.sh.z" columns give the raw and adjusted Z-scores of "pathway cohesion," which compares the observed PC1 magnitude to the magnitudes obtained when the observations for each gene are randomized with respect to cells. When such Z-score is high (e.g., for GO:0008009) then multiple genes within the pathway contribute to the coordinated pattern.

6. We can also test "de novo" gene sets whose expression profiles are well correlated within the given dataset. The following procedure will determine "de novo" gene clusters in the data, and build a background model for the expectation of the gene cluster weighted principal component magnitudes. Note the higher trim values for the clusters, as we want to avoid clusters that are formed by outlier cells.

```
clpca <- pagoda.gene.clusters(varinfo, trim = 7.1/ncol(varinfo$mat),
n.clusters = 50, n.cores = 1, plot = FALSE)
```

7. Now the set of top aspects can be recalculated taking these de novo gene clusters into account.

```
df <- pagoda.top.aspects(pwpca, clpca, return.table = TRUE, plot = FALSE,
z.score = 1.96)
head(df)
##               name npc   n    score         z     adj.z
## 339     GO:0003179   1  10 3.495767 11.108780 10.760218
## 338     GO:0003170   1  10 3.495767 11.108780 10.760218
## 4334 geneCluster.8   1 307 3.397680 13.114746 12.814767
## 3570    GO:0060563   1  12 3.220725 10.643172 10.297292
## 1829    GO:0030426   1  39 3.134488 14.644926 14.338584
## 1302    GO:0014009   1  10 3.105600  9.656705  9.307366
```

8. To view top aspects of transcriptional heterogeneity, we will first obtain information on all the significant aspects. We will also determine the overall cell clustering based on this full pathway-level information.

```
# get full info on the top aspects
tam <- pagoda.top.aspects(pwpca, clpca, n.cells = NULL, z.score =
qnorm(0.01/2, lower.tail = FALSE))
# determine overall cell clustering
hc <- pagoda.cluster.cells(tam, varinfo)
```

9. We can then reduce redundant aspects in two steps. In the first step, we will combine pathways that are driven by the same sets of genes. In the second step we will combine aspects that show similar patterns (i.e., separate the same sets of cells).

```
tamr <- pagoda.reduce.loading.redundancy(tam, pwpca, clpca)
tamr2 <- pagoda.reduce.redundancy(tamr, distance.threshold = 0.9, trim =
0, plot = FALSE)
```

10. We can then view these top aspects in a heatmap (Fig. 7). Indeed, we see a correspondence between out derived cell annotations and the previously published annotations.

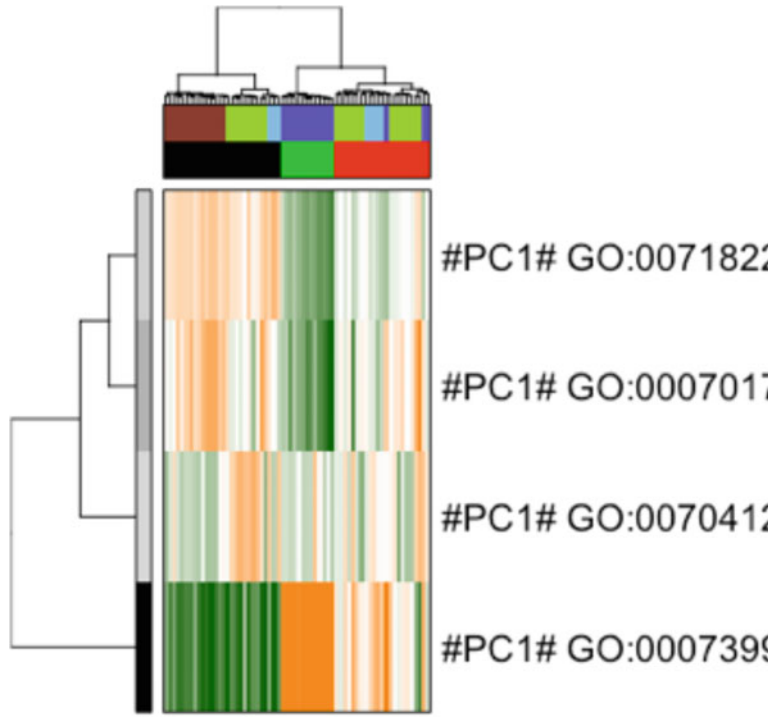

Fig. 7 Pathway expression heatmap for single-cell RNA-seq data from Pollen et al. The columns are cells and the rows represent a cluster of pathways. The row names are assigned to be the top overdispersed aspect in each cluster. The green-to-orange color scheme shows low-to-high weighted PC scores (aspect patterns), where generally orange indicates higher expression and green lower expression. The column colors are cell annotations from the original publication

Fig. 8 Sample screenshot of an interactive PAGODA app

```
col.cols <- rbind(groups = cutree(hc, 3), l2cols)
pagoda.view.aspects(tamr2, cell.clustering = hc, box = TRUE, labCol = NA,
margins = c(0.5, 20), col.cols = rbind(col.cols))
```

11. To interactively browse and explore the output, we can also create a PAGODA app (Fig. 8).

```
# compile a browsable app, showing top three clusters with the top color
bar
app <- make.pagoda.app(tamr2, tam, varinfo, go.env, pwpca, clpca,
col.cols = col.cols, cell.clustering = hc, title = "NPCs")
# show app in the browser (port 1468)
show.app(app, "pollen", browse = TRUE, port = 1468)
```

The PAGODA app allows you to view the gene sets grouped within each aspect (row), as well as genes underlying the detected heterogeneity patterns. In this manner, you can interactively explore the pathways and genes driving each identified transcriptional subpopulation.

Acknowledgment

This work was supported by NIH grant F99CA222750.

References

1. Soneson C, Delorenzi M (2013) A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics 14:91. https://doi.org/10.1186/1471-2105-14-91
2. Jaakkola MK, Seyednasrollah F, Mehmood A, Elo LL (2016) Comparison of methods to detect differentially expressed genes between single-cell populations. Brief Bioinform 18: bbw057. https://doi.org/10.1093/bib/bbw057
3. Kharchenko PV, Silberstein L, Scadden DT (2014) Bayesian approach to single-cell differential expression analysis. Nat Methods 11:740–742. https://doi.org/10.1038/nmeth.2967
4. Finak G, McDavid A, Yajima M, Deng J, Gersuk V, Shalek AK, Slichter CK, Miller HW, McElrath MJ, Prlic M, Linsley PS, Gottardo R (2015) MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in sin gle-cell RNA sequencing data. Genome Biol 16:278. https://doi.org/10.1186/s13059-015-0844-5
5. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102:15545–15550. https://doi.org/10.1073/pnas.0506580102
6. Fan J, Salathia N, Liu R, Kaeser GE, Yung YC, Herman JL, Kaper F, Fan J-B, Zhang K, Chun J, Kharchenko PV (2016) Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. Nat Methods 13:241–244. https://doi.org/10.1038/nmeth.3734
7. Wagner F (2016) The XL-mHG test for enrichment: algorithms, bounds, and power. https://doi.org/10.7287/peerj.preprints.1962v1
8. Huang DW, Sherman BT, Lempicki RA (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res 37:1–13. https://doi.org/10.1093/nar/gkn923
9. R Core Team (2017) R: a language and environment for statistical computing
10. Liberzon A, Birger C, Thorvaldsdóttir H, Ghandi M, Mesirov JP, Tamayo P (2015) The molecular signatures database (MSigDB) hallmark gene set collection. Cell Syst 1:417–425. https://doi.org/10.1016/j.cels.2015.12.004
11. Dunnett CW (1955) A multiple comparison procedure for comparing several treatments with a control. J Am Stat Assoc 50:1096–1121. https://doi.org/10.1080/01621459.1955.10501294
12. Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, Li N, Szpankowski L, Fowler B, Chen P, Ramalingam N, Sun G, Thu M, Norris M, Lebofsky R, Toppani D, Kemp DW, Wong M, Clerkson B, Jones BN, Wu S, Knutsson L, Alvarado B, Wang J, Weaver LS, May AP, Jones RC, Unger MA, Kriegstein AR, West JAA (2014) Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. Nat Biotechnol 32:1053–1058. https://doi.org/10.1038/nbt.2967