scDDboost

Xiuyu Ma

Contents

Abstract:

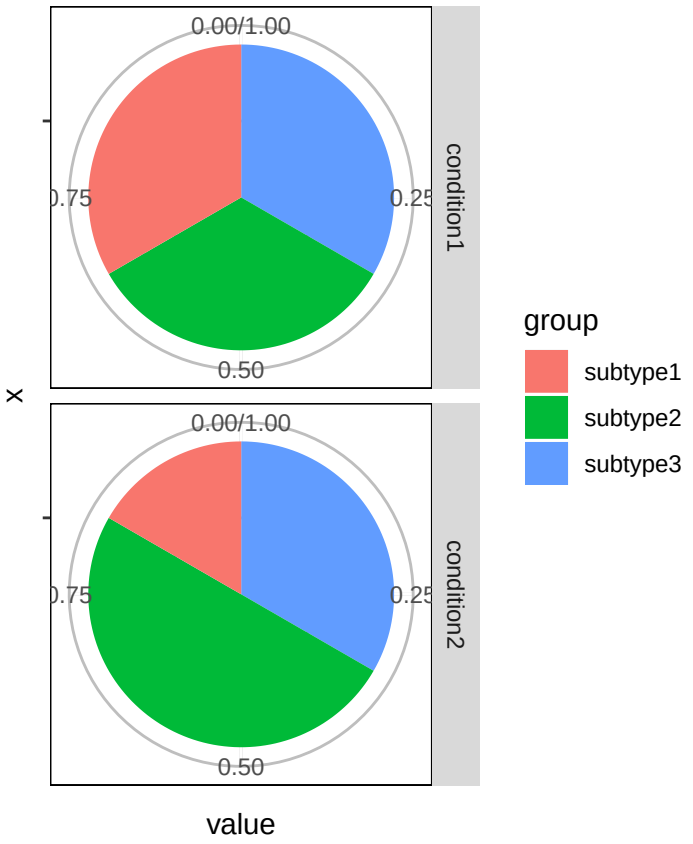
The scDDboost package models single-cell gene expression data (from single-cell RNA-seq) using mixture models in order to explicitly handle heterogeneity within cell populations. In bulk RNA-seq data, where each measurement is an average over thousands of cells, distributions of expression over samples are most often unimodal. In singlecell RNA-seq data, however, even when cells represent genetically homogeneous populations, multimodal distributions of gene expression values over samples are common [1]. This type of heterogeneity is often treated as a nuisance factor in studies of differential expression in single-cell RNA-seq experiments. Here, we explicitly accommodate it in order to improve power to detect differences in expression distributions that are more complicated than a mean shift.

Package scDDboost 1.0.0 Report issues on "github link here"

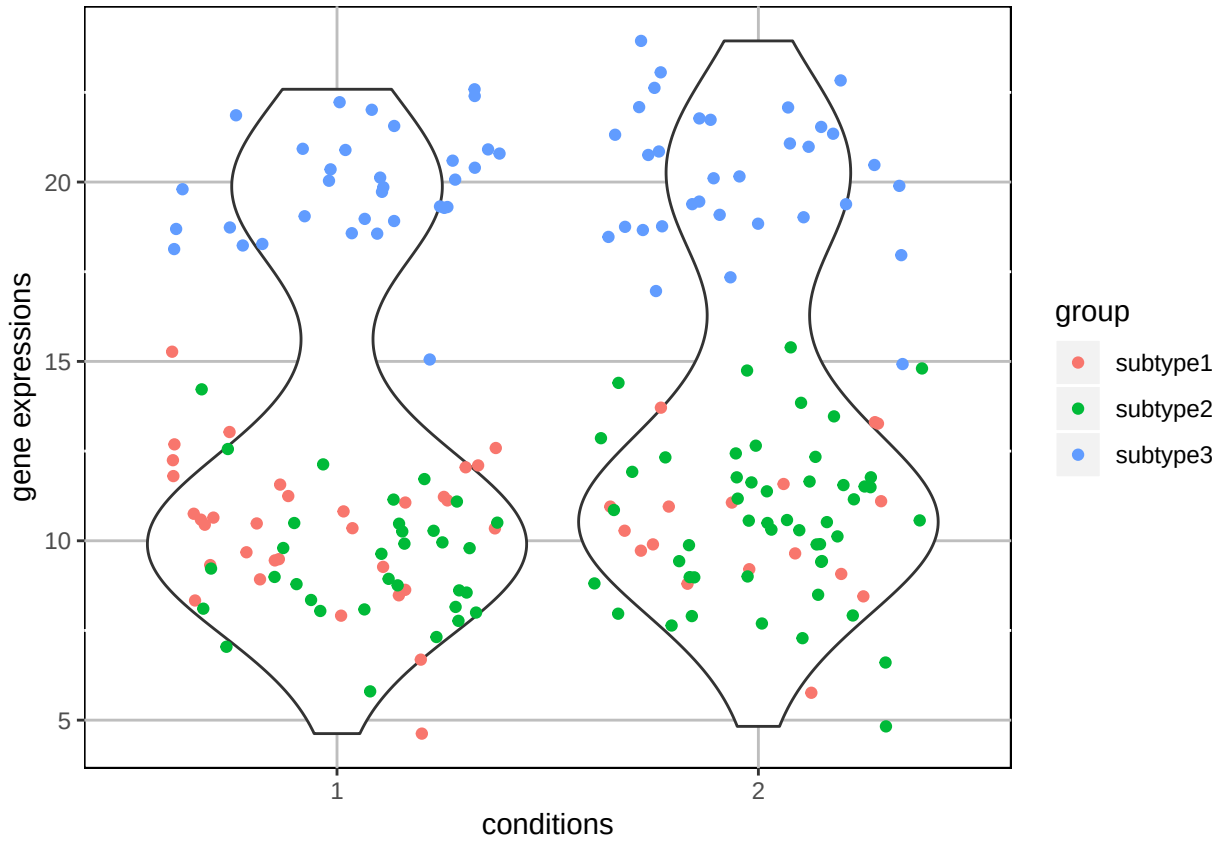
Background 1

Our aim is to identify differential distributed genes across two biological conditions. We view samples as mixture of cells coming from distinct subtypes. Change of proportions of subtypes does not necessarily lead to distributional change of transcripts across conditions, given that subtypes could share lots of genes with equivalent expressed and for two subtypes as long as their total proportion remains unchanged across biological conditions, the change of individual proportions would not lead distributional change at those equivalent expressed genes.

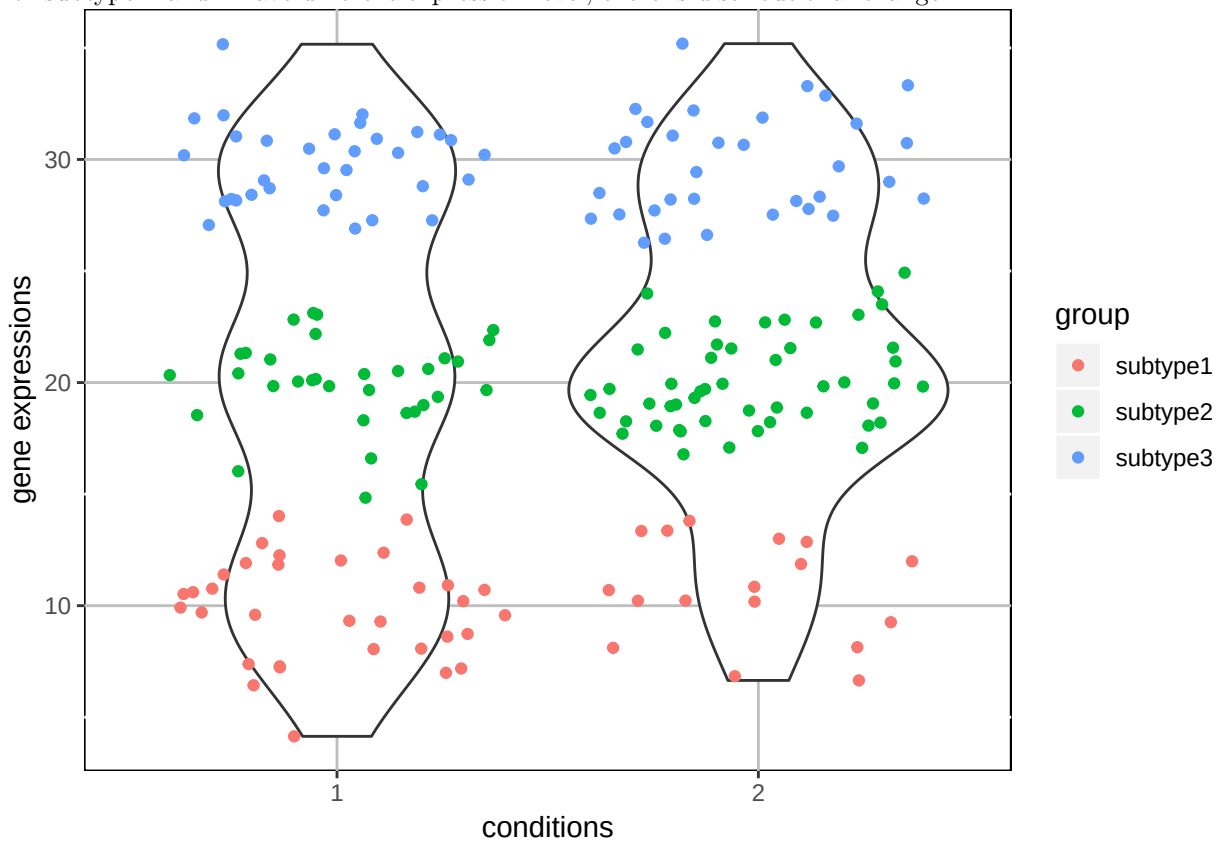
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```



even proportions of subtypes 1 and 2 changed. For gene subtype 1 and 2 have the same expression level, there is no DD.



For subtype 1 and 2 have different expression level, there is distributional change



Identify DD genes 2

In this section, we demonstrate how to cluster cells by modal cluster and how to use the main function **PDD** to find genes with differential distributions
First we need to load the scDDboost package. For each of the following sections in this vignette, we assume this step has been carried out.

```
suppressMessages(library(scDDboost))
```

Next, we load the toy simulated example a *SingleCellExperiment* object that we will use for identifying and classifying DD genes.

```
suppressMessages(data(sim_dat))
```

Verify that this object is a member of the SingleCellExperiment class and that it contains 200 samples and 500 genes. The colData slot (which contains a dataframe of metadata for the cells) should have a column that contains the biological condition or grouping of interest. In this example data, that variable is the 'condition' variable. Note that the input gene set needs to be in SingleCellExperiment format, and should contain normalized counts. In practice, it is also advisable to filter the input gene set to remove genes that have an extremely high proportion of zeroes. A typicall workflow of scDDboost would be following

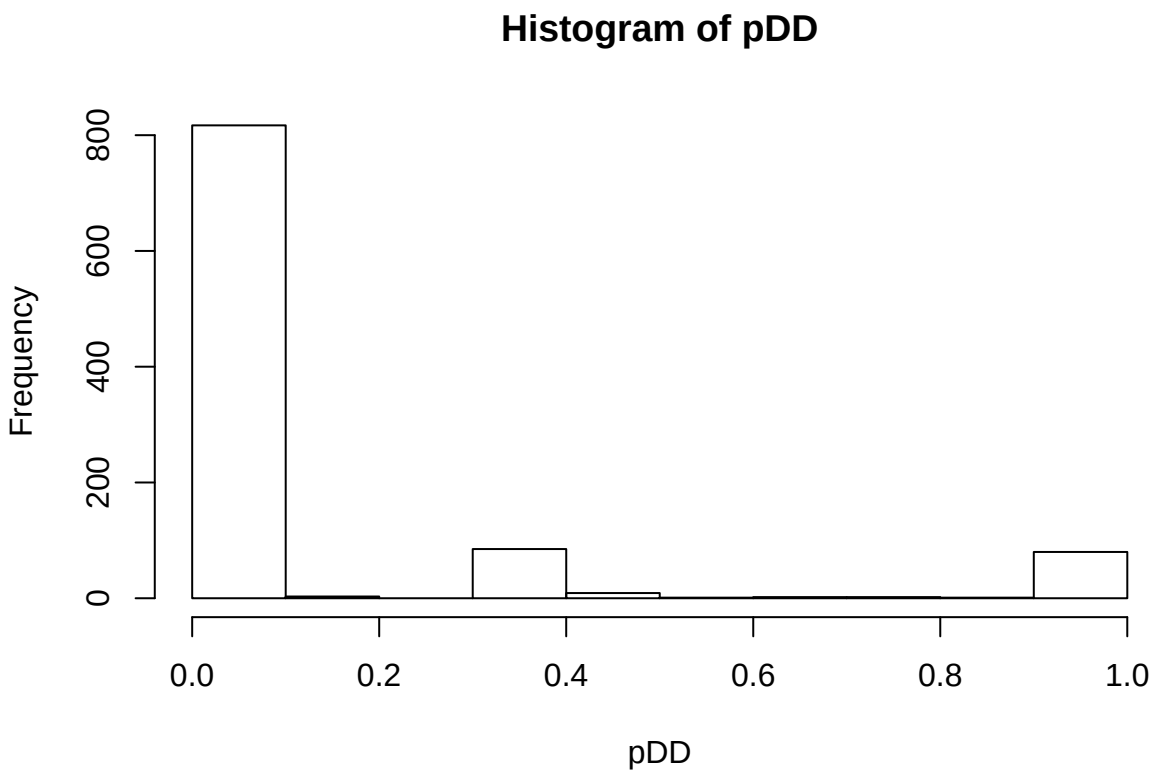
```
suppressMessages(library(SingleCellExperiment))
## expression counts
data_counts = assays(sim_dat)$counts

## condition label
cd = colData(sim_dat)$conditions

## distance matrix
D_c = cal_D(data_counts,4)

## probability of being DD
pDD = PDD(data_counts,cd,2,D_c)
```

```
## 100 genes are all zero counts, not being considered in DD analysis
```

```
## estimated number of subtypes: 4
```

If we set threshold to be 5% then we have estimated DD genes

```
EDD = lsz(pDD,0.05)
hist(pDD)
```



we could also compare to the true label of DD genes

```
DD = which(rowData(sim_dat)$DD > 0)

## proportion of DD genes correctly identified

length(intersect(DD,EDD)) / length(DD)
```

```
## [1] 0.83
```

```
## proportion of DD genes falsely identified

(length(EDD) - length(intersect(DD,EDD))) / length(EDD)
```

```
## [1] 0
```