

# Package ‘TSCAN’

October 12, 2016

**Type** Package

**Title** TSCAN: Tools for Single-Cell ANalysis

**Version** 1.10.2

**Date** 2015-8-15

**Author** Zhicheng Ji, Hongkai Ji

**Maintainer** Zhicheng Ji <zji4@jhu.edu>

**Description** TSCAN enables users to easily construct and tune pseudotemporal cell ordering as well as analyzing differentially expressed genes. TSCAN comes with a user-friendly GUI written in shiny. More features will come in the future.

**License** GPL(>=2)

**Imports** ggplot2, shiny, plyr, grid, fastICA, igraph, combinat, mgcv, mclust, gplots

**VignetteBuilder** knitr

**Suggests** knitr

**Depends** R(>= 2.10.0)

**biocViews** GeneExpression, Visualization, GUI

**NeedsCompilation** no

## R topics documented:

difftest . . . . .	2
exprmclust . . . . .	3
lpsdata . . . . .	4
orderscore . . . . .	4
plotmclust . . . . .	5
preprocess . . . . .	6
singlegeneplot . . . . .	8
TSCAN . . . . .	9
TSCANorder . . . . .	9
TSCANui . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

difftest

*difftest*

---

**Description**

testing differentially expressed genes

**Usage**

```
difftest(data, TSCANorder, df = 3)
```

**Arguments**

data	The raw single_cell data, which is a numeric matrix or data.frame. Rows represent genes/features and columns represent single cells.
TSCANorder	The TSCAN ordering generated by function <a href="#">TSCANorder</a> .
df	Numeric value specifying the degree of freedom used in the GAM model.

**Details**

This function tests whether a gene is significantly expressed given pseudotime ordering. Likelihood ratio test is performed to compare a generalized additive model (GAM) with a constant fit to get the p-values. The p-values are adjusted for multiple testing by *fdr*.

**Value**

Data frame containing pvalues and qvalues of testing differentially expression.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
data(lpsdata)
procdata <- preprocess(lpsdata)
lpsorder <- TSCANorder(exprmclust(procdata))
diffval <- difftest(procdata,lpsorder)
#Selected differentially expressed genes under qvlue cutoff of 0.05
row.names(diffval)[diffval$qval < 0.05]
```

---

`exprmclust`*exprmclust*

---

**Description**

Perform model-based clustering on expression values

**Usage**

```
exprmclust(data, clusternum = 2:9, modelNames = "VVV", reduce = T)
```

**Arguments**

<code>data</code>	The raw <code>single_cell</code> data, which is a numeric matrix or <code>data.frame</code> . Rows represent genes/features and columns represent single cells.
<code>clusternum</code>	An integer vector specifying all possible cluster numbers. The best cluster number will be picked using BIC. The minimum value should be two other
<code>modelNames</code>	model to be used in model-based clustering. By default "ellipsoidal, varying volume, shape, and orientation" is used.
<code>reduce</code>	Whether to perform the PCA on the expression data.

**Details**

By default, this function first uses principal component analysis (PCA) to reduce dimensionality of original data. It then performs model-based clustering on the transformed expression values. A minimum-spanning-tree is constructed to link the cluster centers. The clustering results will be used for TSCAN ordering.

**Value**

if more than one cluster detected, a list containing

- `pcareduces` Numeric matrix containing the transformed expression values after PCA.
- `MSTtree` `igraph` object which is the result of constructing MST.
- `clusterid` A named vector specifying which cluster the cells belong to.
- `clucenter` Numeric matrix of the cluster centers.

if only one cluster detected, a list containing

- `pcareduces` Numeric matrix containing the transformed expression values after PCA.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

## References

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458), 611-631.

## Examples

```
data(lpsdata)
procdata <- preprocess(lpsdata)
exprmclust(procdata)
```

---

lpsdata	<i>Single-cell RNA-seq data for BMDC cells before and after LPS stimulation</i>
---------	---

---

## Description

The dataset contains 16776 rows and 131 columns. Each row represent a gene and each column represent a single cell. This dataset is a subset of single-cell RNA-seq data provided by GEO GSE48968. Only unstimulated cells and cells after 6h of LPS stimulation are retained for the purpose of demonstration. Genes which have raw expression values of greater than zero in at least one cell are retained. For the original dataset please refer to GSE48968 on GEO (<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE48968>).

## Format

A matrix with 16776 rows and 131 variables

## Source

<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE48968>

## References

Shalek, A. K., Satija, R., Shuga, J., Trombetta, J. J., Gennert, D., Lu, D., ... & Regev, A. (2014). Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*.

---

orderscore	<i>orderscore</i>
------------	-------------------

---

## Description

Calculate pseudotemporal ordering scores for orders

## Usage

```
orderscore(subpopulation, orders)
```

**Arguments**

- subpopulation    Data frame with two columns. First column: cell names. Second column: sub-population codes.
- orders            A list with various length containing pseudotime orderings.

**Details**

This function calculates pseudotemporal ordering scores (POS) based on the sub-population information and order information given by users. Cells should come from at least two cell sub-populations. These sub-population should be coded as 0,1,2,...

**Value**

a numeric vector of calculated POS.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
data(lpsdata)
procdata <- preprocess(lpsdata)
subpopulation <- data.frame(cell = colnames(procdata), sub = ifelse(grepl("Unstimulated", colnames(procdata)), 0, 1))
lpsmclust <- exprmclust(procdata)
#Comparing default TSCAN ordering and tuned TSCAN ordering
order1 <- TSCANorder(lpsmclust)
order2 <- TSCANorder(lpsmclust, c(1,2,3))
orders <- list(order1,order2)
orderscore(subpopulation, orders)
```

---

plotmclust

*plotmclust*

---

**Description**

Plot the model-based clustering results

**Usage**

```
plotmclust(mclustobj, x = 1, y = 2, MSTorder = NULL, show_tree = T,
           show_cell_names = T, cell_name_size = 3, markerexpr = NULL)
```

**Arguments**

<code>mclustobj</code>	The exact output of <code>exprmclust</code> function.
<code>x</code>	The column of data after dimension reduction to be plotted on the horizontal axis.
<code>y</code>	The column of data after dimension reduction to be plotted on the vertical axis.
<code>MSTorder</code>	The arbitrary order of cluster to be shown on the plot.
<code>show_tree</code>	Whether to show the links between cells connected in the minimum spanning tree.
<code>show_cell_names</code>	Whether to draw the name of each cell in the plot.
<code>cell_name_size</code>	The size of cell name labels if <code>show_cell_names</code> is TRUE.
<code>markerexpr</code>	The gene expression used to define the size of nodes.

**Details**

This function will plot the gene expression data after dimension reduction and show the clustering results.

**Value**

A `ggplot2` object.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
data(lpsdata)
procddata <- preprocess(lpsdata)
lpsmclust <- exprmclust(procddata)
plotmclust(lpsmclust)
```

---

```
preprocess
```

```
preprocess
```

---

**Description**

preprocess the raw single-cell data

**Usage**

```
preprocess(data, clusternum = NULL, takelog = TRUE, logbase = 2,
  pseudocount = 1, minexpr_value = 1, minexpr_percent = 0.5,
  cvcutoff = 1)
```

**Arguments**

<code>data</code>	The raw <code>single_cell</code> data, which is a numeric matrix or <code>data.frame</code> . Rows represent genes/features and columns represent single cells.
<code>clusternum</code>	The number of clusters for doing cluster, typically 5 percent of number of all genes. The clustering will be done after all the transformation and trimming. If NULL no clustering will be performed.
<code>takelog</code>	Logical value indicating whether to take logarithm
<code>logbase</code>	Numeric value specifying base of logarithm
<code>pseudocount</code>	Numeric value to be added to the raw data when taking logarithm
<code>minexpr_value</code>	Numeric value specifying the minimum cutoff of log transformed (if <code>takelog</code> is TRUE) value
<code>minexpr_percent</code>	Numeric value specifying the lowest percentage of highly expressed cells (expression value bigger than <code>minexpr_value</code> ) for the genes/features to be retained.
<code>cvcutoff</code>	Numeric value specifying the minimum value of coefficient of variance for the genes/features to be retained.

**Details**

This function first takes logarithm of the raw data and then filters out genes/features in which too many cells are low expressed. It also filters out genes/features with low coefficient of variance which indicates the genes/features does not contain much information. The default setting will first take  $\log_2$  of the raw data after adding a pseudocount of 1. Then genes/features in which at least half of cells have expression values are greater than 1 and the coefficients of variance across all cells are at least 1 are retained.

**Value**

Matrix or data frame with the same format as the input dataset.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
data(lpsdata)
procdata <- preprocess(lpsdata)
```

singlegeneplot      *singlegeneplot*

---

### Description

plot expression values of individual genes against pseudotime axis

### Usage

```
singlegeneplot(geneexpr, TSCANorder, cell_size = 2)
```

### Arguments

geneexpr	The gene expression values. Names should agree with the pseudotime information.
TSCANorder	The output of function <a href="#">TSCANorder</a> .
cell_size	Size of cells in the plot.

### Details

This function plots the expression values of individual genes against given pseudotime

### Value

ggplot2 object.

### Author(s)

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

### Examples

```
data(lpsdata)
procdata <- preprocess(lpsdata)
lpsmclust <- exprmclust(procdata)
lpsorder <- TSCANorder(lpsmclust, orderonly=FALSE, flip=TRUE)
#Choose STAT1 gene expression to plot
STAT2expr <- log2(lpsdata["STAT2",]+1)
singlegeneplot(STAT2expr, lpsorder)
```



---

TSCAN

*TSCAN: Tools for Single-Cell ANalysis*


---

**Description**

This package provides essential tools used in analyzing data from single-cell experiments

**Details**

TSCAN enables users to easily construct and tune pseudotemporal cell ordering as well as analyzing differentially expressed genes. TSCAN comes with a user-friendly GUI written in shiny. More functions will come in the future.

---

TSCANorder

*TSCANorder*


---

**Description**

Construct TSCAN order after `exprmclust`

**Usage**

```
TSCANorder(mclustobj, MSTorder = NULL, orderonly = T, flip = F,
            listbranch = F)
```

**Arguments**

<code>mclustobj</code>	The exact output of the <code>exprmclust</code> function.
<code>MSTorder</code>	A numeric vector specifying the order of clusters.
<code>orderonly</code>	Only return the ordering. State or pseudotime information will not be returned
<code>flip</code>	whether to flip the ordering
<code>listbranch</code>	whether to list the ordering results of all possible branches

**Details**

This function takes the exact output of `exprmclust` function and construct TSCAN order by mapping all cells onto the path that connects cluster centers. Users can also specify their own path.

**Value**

if `orderonly = F`, a vector of ordered cell names. if `orderonly = T`, a data frame of ordered cell names, cell states and pseudotime.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
data(lpsdata)
procdata <- preprocess(lpsdata)
lpsmclust <- exprmclust(procdata)
TSCANorder(lpsmclust)
```

---

TSCANui

*TSCANui*

---

**Description**

Launch the TSCAN user interface in local machine

**Usage**

```
TSCANui()
```

**Details**

This function will automatically launch the TSCAN user interface in a web browser. The user interface provides many powerful functions which is not available by command line programming. It also provides a much easier and more convenient way to quickly explore single cell data and construct pseudotime analysis. The user interface can also be accessed by <http://zhiji.shinyapps.io/TSCAN>. Neither R nor any packages are required in this online version. However, it is highly recommended that the user interface be launched locally for faster running speed.

**Author(s)**

Zhicheng Ji, Hongkai Ji <zji4@zji4.edu>

**Examples**

```
## Not run:
  TSCANui()

## End(Not run)
```

# Index

`difftest`, [2](#)

`exprmclust`, [3](#), [6](#), [9](#)

`lpsdata`, [4](#)

`orderscore`, [4](#)

`plotmclust`, [5](#)

`preprocess`, [6](#)

`singlegeneplot`, [8](#)

`TSCAN`, [9](#)

`TSCAN-package (TSCAN)`, [9](#)

`TSCANorder`, [2](#), [8](#), [9](#)

`TSCANui`, [10](#)