Package ‘monocle’

March 10, 2014

Type Package

Title Analysis tools for single-cell expression experiments.

Version 0.99.0

Date 2013-11-19

Author Cole Trapnell

Maintainer Cole Trapnell <cole@broadinstitute.org>

Description Monocle performs differential expression and time-series analysis for single-cell expression experiments. It orders individual cells according to progress through a biological process, without knowing ahead of time which genes define progress through that process. Monocle also performs differential expression analysis, clustering, visualization, and other useful tasks on single cell expression data. It is designed to work with RNA-Seq and qPCR data, but could be used with other types as well.

License Artistic-2.0

Depends Biobase, VGAM, irlba, matrixStats, igraph, combinat, fastICA, grid, ggplot2, reshape2, plyr, parallel, methods

VignetteBuilder knitr

Suggests knitr

LazyData true

R topics documented:

monocle-package ........................................ 2
CellDataSet ............................................. 3
cellPairwiseDistances ................................. 3
cellPairwiseDistances<- ................................ 4
compareModels ......................................... 5
detectGenes ............................................ 5
differentialGeneTest ................................. 6
monocle-package

Differential expression and time-series analysis for single-cell expression experiments.

Description

Monocle is a set of tools for analyzing single-cell gene expression experiments. Monocle has been tested with RNA-Seq and qPCR, but could work with other types of data as well. The package provides basic differential expression and clustering tools, as well as more sophisticated tools. Monocle can place individual cells in the order of their progress through a biological process such as cell differentiation. This enables Monocle to find genes that are differentially expressed as a function of that process.

Details

Package: monocle
Type: Package
Version: 0.1.0
Date: 2013-11-19
License: Boost
CellDataSet

Author(s)
Cole Trapnell Maintainer: Cole Trapnell <cole@broadinstitute.org>

References
Trapnell, Cacchiarelli, et al

CellDataSet  

The CellDataSet class

Description
The main class used by Monocle to hold single cell expression data. CellDataSet extends the basic Bioconductor ExpressionSet class.

Details
This class is initialized from a matrix of expression values. Methods that operate on CellDataSet objects constitute the basic Monocle workflow.

Slots
- reducedDimS: Matrix of class "numeric", containing the source values computed by Independent Components Analysis.
- reducedDimW: Matrix of class "numeric", containing the whitened expression values computed during Independent Components Analysis.
- reducedDimA: Matrix of class "numeric", containing the weight values computed by Independent Components Analysis.
- minSpanningTree: Object of class "igraph", containing the minimum spanning tree used by Monocle to order cells according to progress through a biological process.
- cellPairwiseDistances: Matrix of class "numeric", containing the pairwise distances between cells in the reduced dimension space.

cellPairwiseDistances

Retrieves a matrix capturing distances between each cell in the reduced-dimensionality space

Description
Retrieves a matrix capturing distances between each cell in the reduced-dimensionality space

Usage

cellPairwiseDistances(cds)
**cellPairwiseDistances**

Sets the matrix containing distances between each pair of cells used by Monocle during cell ordering. Not intended to be called directly.

**Arguments**

- `cds` expression data matrix for an experiment

**Value**

A square, symmetric matrix containing the distances between each cell in the reduced-dimensionality space.

**Examples**

```r
data(HSMM)
D <- cellPairwiseDistances(HSMM)
```

**Description**

Sets the matrix containing distances between each pair of cells used by Monocle during cell ordering. Not intended to be called directly.

**Usage**

```r
cellPairwiseDistances(cds) <- value
```

**Arguments**

- `cds` A CellDataSet object.
- `value` a square, symmetric matrix containing pairwise distances between cells.

**Value**

An updated CellDataSet object
compareModels  Compare model fits

Description
Performs likelihood ratio tests on nested vector generalized additive models

Usage
```r
compareModels(full_models, reduced_models)
```

Arguments
- `full_models`: a list of models, e.g. as returned by `fitModels()`, forming the numerators of the L.R.Ts.
- `reduced_models`: a list of models, e.g. as returned by `fitModels()`, forming the denominators of the L.R.Ts.

Value
a data frame containing the p values and q-values from the likelihood ratio tests on the parallel arrays of models.

detectGenes  Sets the global expression detection threshold to be used with this CellDataSet. Counts how many cells each feature in a CellDataSet object that are detectably expressed above a minimum threshold. Also counts the number of genes above this threshold are detectable in each cell.

Description
Sets the global expression detection threshold to be used with this CellDataSet. Counts how many cells each feature in a CellDataSet object that are detectably expressed above a minimum threshold. Also counts the number of genes above this threshold are detectable in each cell.

Usage
```r
detectGenes(cds, min_expr)
```

Arguments
- `cds`: the CellDataSet upon which to perform this operation
- `min_expr`: the expression threshold
differentialGeneTest

Tests each gene for differential expression as a function of progress through a biological process, or according to other covariates as specified.

Description

Tests each gene for differential expression as a function of progress through a biological process, or according to other covariates as specified.

Usage

differentialGeneTest(cds,
  fullModelFormulaStr = "expression~VGAM::bs(Pseudotime, df=3)",
  reducedModelFormulaStr = "expression~1", min_expr = 0.1, max_expr = Inf,
  cores = 1)

Arguments

cds a CellDataSet object upon which to perform this operation
fullModelFormulaStr a formula string specifying the full model in differential expression tests (i.e. likelihood ratio tests) for each gene/feature.
reducedModelFormulaStr a formula string specifying the reduced model in differential expression tests (i.e. likelihood ratio tests) for each gene/feature.
min_expr the minimum expression value considered reliably measured in the experiment
max_expr the maximum expression value considered reliably measured in the experiment
cores the number of cores to be used while testing each gene for differential expression

Value

a data frame containing the p values and q-values from the likelihood ratio tests on the parallel arrays of models.
fitModel  

Fits a model for each gene in a CellDataSet object.

Description

Fits a model for each gene in a CellDataSet object.

Usage

```r
fitModel(cds, modelFormulaStr = "expression~VGAM::bs(Pseudotime, df=3)",
          min_expr = 0.1, max_expr = Inf, cores = 1)
```

Arguments

- `cds`: the CellDataSet upon which to perform this operation
- `modelFormulaStr`: a formula string specifying the model to fit for the genes.
- `min_expr`: the minimum expression value considered reliably measured in the experiment.
- `max_expr`: the maximum expression value considered reliably measured in the experiment.
- `cores`: the number of processor cores to be used during fitting.

Details

This function fits a Tobit-family vector generalized additive model (VGAM) from the VGAM package for each gene in a CellDataSet. The default formula string specifies that the (log transformed) expression values follow a Tobit distribution with upper and lower bounds specified by `max_expr` and `min_expr`, respectively. By default, expression levels are modeled as smooth functions of the Pseudotime value of each cell. That is, expression is a function of progress through the biological process. More complicated formulae can be provided to account for additional covariates (e.g. day collected, genotype of cells, media conditions, etc).

Value

A list of VGAM model objects.

HSMM  
The HSMM single-cell RNA-Seq timeseries from Trapnell, Cacchiarelli et al.

Description

The HSMM single-cell RNA-Seq timeseries from Trapnell, Cacchiarelli et al.
Author(s)

Cole Trapnell <cole@broadinstitute.org>

References


minSpanningTree<-  Retrieves the minimum spanning tree generated by Monocle during cell ordering.

Description

Retrieves the minimum spanning tree generated by Monocle during cell ordering.

Usage

minSpanningTree(cds)

Arguments

cds  expression data matrix for an experiment

Value

An igraph object representing the CellDataSet’s minimum spanning tree.

Examples

data(HSMM)
T <- minSpanningTree(HSMM)

minSpanningTree<-  Sets the minimum spanning tree used by Monocle during cell ordering. Not intended to be called directly.

Description

Sets the minimum spanning tree used by Monocle during cell ordering. Not intended to be called directly.

Usage

minSpanningTree(cds) <- value
newCellDataSet

Arguments

cds A CellDataSet object.
value an igraph object describing the minimum spanning tree.

Value

An updated CellDataSet object

newCellDataSet  Creates a new CellDataSet object.

Description

Creates a new CellDataSet object.

Usage

newCellDataSet(cellData, phenoData = NULL, featureData = NULL)

Arguments

cellData expression data matrix for an experiment
phenoData data frame containing attributes of individual cells
featureData data frame containing attributes of features (e.g. genes)

Value

a new CellDataSet object

Examples

## Not run:
sample_sheet_small <- read.delim("../data/sample_sheet_small.txt", row.names=1)
sample_sheet_small$Time <- as.factor(sample_sheet_small$Time)
gene_annotations_small <- read.delim("../data/gene_annotations_small.txt", row.names=1)
fpkm_matrix_small <- read.delim("../data/fpkm_matrix_small.txt")
pd <- new("AnnotatedDataFrame", data = sample_sheet_small)
fd <- new("AnnotatedDataFrame", data = gene_annotations_small)
HSMM <- new("CellDataSet", exprs = as.matrix(fpkm_matrix_small), phenoData = pd, featureData = fd)

## End(Not run)
**orderCells**

*Orders cells according to progress through a learned biological process.*

**Description**

Orders cells according to progress through a learned biological process.

**Usage**

```
orderCells(cds, num_paths = 1, reverse = FALSE)
```

**Arguments**

- `cds` 
  the CellDataSet upon which to perform this operation
- `num_paths` 
  the number of end-point cell states to allow in the biological process.
- `reverse` 
  whether to reverse the beginning and end points of the learned biological process.

**Value**

an updated CellDataSet object, in which phenoData contains values for State and Pseudotime for each cell

---

**plot_genes_in_pseudotime**

*Plots expression for one or more genes as a function of pseudotime*

**Description**

Plots expression for one or more genes as a function of pseudotime

**Usage**

```
plot_genes_in_pseudotime(cds_subset, min_expr = 0.1, cell_size = 0.75, nrow = NULL, ncol = 1, panel_order = NULL, color_by = "State")
```

**Arguments**

- `cds_subset` 
  CellDataSet for the experiment
- `min_expr` 
  the minimum (untransformed) expression level to use in plotted the genes.
- `cell_size` 
  the size (in points) of each cell used in the plot
- `nrow` 
  the number of rows used when laying out the panels for each gene’s expression
- `ncol` 
  the number of columns used when laying out the panels for each gene’s expression
- `panel_order` 
  the order in which genes should be layed out (left-to-right, top-to-bottom)
- `color_by` 
  the cell attribute (e.g. the column of pData(cds)) to be used to color each cell
plot_genes_jitter

Value

a ggplot2 plot object

Examples

```r
## Not run:
data(HSMM)
my_genes <- row.names(subset(fData(HSMM), gene_short_name %in% c("CDK1", "MEF2C", "MYH3")))
cds_subset <- HSMM[my_genes,]
plot_genes_in_pseudotime(cds_subset, color_by="Time")
## End(Not run)
```

Description

Plots expression for one or more genes as a jittered, grouped points

Usage

```r
plot_genes_jitter(cds_subset, grouping = "State", min_expr = 0.1,
                   cell_size = 0.75, nrow = NULL, ncol = 1, panel_order = NULL,
                   color_by = NULL, plot_trend = F)
```

Arguments

- `cds_subset`: CellDataSet for the experiment
- `grouping`: the cell attribute (e.g. the column of pData(cds)) to group cells by on the horizontal axis
- `min_expr`: the minimum (untransformed) expression level to use in plotted the genes.
- `cell_size`: the size (in points) of each cell used in the plot
- `nrow`: the number of rows used when laying out the panels for each gene’s expression
- `ncol`: the number of columns used when laying out the panels for each gene’s expression
- `panel_order`: the order in which genes should be layed out (left-to-right, top-to-bottom)
- `color_by`: the cell attribute (e.g. the column of pData(cds)) to be used to color each cell
- `plot_trend`: whether to plot a trendline tracking the average expression across the horizontal axis.

Value

a ggplot2 plot object
plot_spanning_tree

Plots the minimum spanning tree on cells.

Description

Plots the minimum spanning tree on cells.

Usage

plot_spanning_tree(cds, x = 1, y = 2, color_by = "State", show_tree = T, show_backbone = T, backbone_color = "black", marker = NULL)

Arguments

cds          CellDataSet for the experiment
x            the column of reducedDimS(cds) to plot on the horizontal axis
y            the column of reducedDimS(cds) to plot on the vertical axis
color_by     the cell attribute (e.g. the column of pData(cds)) to map to each cell’s color
show_tree    whether to show the links between cells connected in the minimum spanning tree
show_backbone whether to show the diameter path of the MST used to order the cells
backbone_color the color used to render the backbone.
marker       a gene name or gene id to use for setting the size of each cell in the plot

Value

a ggplot2 plot object

Examples

## Not run:
data(HSMM)
plot_spanning_tree(HSMM)
plot_spanning_tree(HSMM, color_by="Pseudotime", show_backbone=F)
plot_spanning_tree(HSMM, marker="MYH3")

## End(Not run)
reducedDimA

Retrieves the weights that transform the cells’ coordinates in the reduced dimension space back to the full (whitened) space.

Description

Retrieves the weights that transform the cells’ coordinates in the reduced dimension space back to the full (whitened) space.

Usage

reducedDimA(cds)

Arguments

cds A CellDataSet object.

Value

A matrix that when multiplied by a reduced-dimension set of coordinates for the CellDataSet, recovers a matrix in the full (whitened) space.

Examples

data(HSMM)
A <- reducedDimA(HSMM)

reducedDimA<-

Sets the weights transform the cells’ coordinates in the reduced dimension space back to the full (whitened) space.

Description

Sets the weights transform the cells’ coordinates in the reduced dimension space back to the full (whitened) space.

Usage

reducedDimA(cds) <- value

Arguments

cds A CellDataSet object.
value A whitened expression data matrix

Value

An updated CellDataSet object
reducedDimS<-  

Retrieves the coordinates of each cell in the reduced-dimensionality space generated by calls to reduceDimension.

Description
Retrieves the coordinates of each cell in the reduced-dimensionality space generated by calls to reduceDimension.

Usage
reducedDimS(cds)

Arguments
cds  A CellDataSet object.

Value
A matrix, where rows are cell coordinates and columns correspond to dimensions of the reduced space.

Examples
data(HSMM)
S <- reducedDimS(HSMM)

reducedDimS<-  

Sets the coordinates of each cell in the reduced-dimensionality space. Not intended to be called directly.

Description
Sets the coordinates of each cell in the reduced-dimensionality space. Not intended to be called directly.

Usage
reducedDimS(cds) <- value

Arguments
cds  A CellDataSet object.
value  A matrix of coordinates specifying each cell’s position in the reduced-dimensionality space.
reducedDimW

Value
An update CellDataSet object

reducedDimW retrieves the expression values for each cell (as a matrix) after whitening during independent component analysis.

Description
Retrieves the expression values for each cell (as a matrix) after whitening during independent component analysis.

Usage
reducedDimW(cds)

Arguments
cds
A CellDataSet object.

Value
A matrix, where each row is a set of whitened expression values for a feature and columns are cells.

Examples
data(HSMM)
W <- reducedDimW(HSMM)

reducedDimW<- sets the whitened expression values for each cell prior to independent component analysis. Not intended to be called directly.

Description
Sets the whitened expression values for each cell prior to independent component analysis. Not intended to be called directly.

Usage
reducedDimW(cds) <- value

Arguments
cds
A CellDataSet object.
value
A whitened expression data matrix
### reduceDimension

**Description**

Computes a projection of a CellDataSet object into a lower dimensional space

**Usage**

```r
reduceDimension(cds, max_components = 2, use_irlba = TRUE)
```

**Arguments**

- `cds`: the CellDataSet upon which to perform this operation
- `max_components`: the dimensionality of the reduced space
- `use_irlba`: Whether to use the IRLBA package for ICA reduction.

**Details**

Currently, Monocle supports dimensionality reduction with Independent Component Analysis (ICA).

**Value**

An updated CellDataSet object

---

### responseMatrix

**Description**

Generates a matrix of response values for a set of fitted models

**Usage**

```r
responseMatrix(models)
```

**Arguments**

- `models`: a list of models, e.g. as returned by `fitModels()`
setOrderingFilter

Value

A matrix where each row is a vector of response values for a particular feature’s model, and columns are cells.

Description

Sets the features (e.g. genes) to be used for ordering cells in pseudo-time.

Usage

setOrderingFilter(cds, ordering_genes)

Arguments

cds the CellDataSet upon which to perform this operation
ordering_genes a vector of feature ids (from the CellDataSet’s featureData) used for ordering cells

Value

an updated CellDataSet object
Index

*Topic data
    HSMM, 7

*Topic package
    monocle-package, 2

CellDataSet, 3
CellDataSet-class(CellDataSet), 3
cellPairwiseDistances, 3
cellPairwiseDistances<-, 4
compareModels, 5
detectGenes, 5
differentialGeneTest, 6
fitModel, 7
HSMM, 7
minSpanningTree, 8
minSpanningTree<-, 8
monocle(monocle-package), 2
monocle-package, 2
newCellDataSet, 9
orderCells, 10
plot_genes_in_pseudotime, 10
plot_genes_jitter, 11
plot_spanning_tree, 12
reducedDimA, 13
reducedDimA<-, 13
reducedDimS, 14
reducedDimS<-, 14
reducedDimW, 15
reducedDimW<-, 15
reduceDimension, 16
responseMatrix, 16
setOrderingFilter, 17