

Package ‘Rmagic’

November 21, 2019

Type Package

Title MAGIC - Markov Affinity-Based Graph Imputation of Cells

Version 2.0.3

Maintainer Scott Gigante <scott.gigante@yale.edu>

Description MAGIC (Markov affinity-based graph imputation of cells) is a method for addressing technical noise in single-cell data, including under-sampling of mRNA molecules, often termed “dropout” which can severely obscure important gene-gene relationships. MAGIC shares information across similar cells, via data diffusion, to denoise the cell count matrix and fill in missing transcripts. Read more: van Dijk et al. (2018) <[DOI:10.1016/j.cell.2018.05.061](https://doi.org/10.1016/j.cell.2018.05.061)>.

Depends R (>= 3.3), Matrix (>= 1.2-0)

Imports methods, stats, reticulate (>= 1.4), ggplot2

Suggests Seurat (>= 3.0.0), readr, viridis, phateR

License GPL-2 | file LICENSE

LazyData true

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author David van Dijk [aut],
Scott Gigante [cre] (<<https://orcid.org/0000-0002-4544-2764>>)

Repository CRAN

Date/Publication 2019-11-21 15:30:02 UTC

R topics documented:

<code>as.data.frame.magic</code>	2
<code>as.matrix.magic</code>	2
<code>ggplot.magic</code>	3
<code>install.magic</code>	3
<code>library.size.normalize</code>	4
<code>magic</code>	5

magic_testdata	7
print.magic	8
pymagic_is_available	8
summary.magic	9

Index 10

as.data.frame.magic *Convert a MAGIC object to a data.frame*

Description

Returns the smoothed data matrix

Usage

```
## S3 method for class 'magic'
as.data.frame(x, ...)
```

Arguments

x	A fitted MAGIC object
...	Arguments for as.data.frame()

as.matrix.magic *Convert a MAGIC object to a matrix*

Description

Returns the smoothed data matrix

Usage

```
## S3 method for class 'magic'
as.matrix(x, ...)
```

Arguments

x	A fitted MAGIC object
...	Arguments for as.matrix()

ggplot.magic	<i>Convert a MAGIC object to a data.frame for ggplot</i>
--------------	--

Description

Passes the smoothed data matrix to ggplot

Usage

```
## S3 method for class 'magic'  
ggplot(data, ...)
```

Arguments

data	A fitted MAGIC object
...	Arguments for ggplot()

Examples

```
if (pymagic_is_available() && require(ggplot2)) {  
  
  data(magic_testdata)  
  data_magic <- magic(magic_testdata, genes=c("VIM", "CDH1", "ZEB1"))  
  ggplot(data_magic, aes(VIM, CDH1, colour=ZEB1)) +  
    geom_point()  
  
}
```

install.magic	<i>Install MAGIC Python Package</i>
---------------	-------------------------------------

Description

Install MAGIC Python package into a virtualenv or conda env.

Usage

```
install.magic(envname = "r-reticulate", method = "auto",  
             conda = "auto", pip = TRUE, ...)
```

Arguments

envname	Name of environment to install packages into
method	Installation method. By default, "auto" automatically finds a method that will work in the local environment. Change the default to force a specific installation method. Note that the "virtualenv" method is not available on Windows.
conda	Path to conda executable (or "auto" to find conda using the PATH and other conventional install locations).
pip	Install from pip, if possible.
...	Additional arguments passed to conda_install() or virtualenv_install().

Details

On Linux and OS X the "virtualenv" method will be used by default ("conda" will be used if virtualenv isn't available). On Windows, the "conda" method is always used.

library.size.normalize

Performs L1 normalization on input data such that the sum of expression values for each cell sums to 1, then returns normalized matrix to the metric space using median UMI count per cell effectively scaling all cells as if they were sampled evenly.

Description

Performs L1 normalization on input data such that the sum of expression values for each cell sums to 1, then returns normalized matrix to the metric space using median UMI count per cell effectively scaling all cells as if they were sampled evenly.

Usage

```
library.size.normalize(data, verbose = FALSE)
```

Arguments

data	matrix (n_samples, n_dimensions) 2 dimensional input data array with n cells and p dimensions
verbose	boolean, default=FALSE. If true, print verbose output

Value

data_norm matrix (n_samples, n_dimensions) 2 dimensional array with normalized gene expression values

magic	<i>Perform MAGIC on a data matrix</i>
-------	---------------------------------------

Description

Markov Affinity-based Graph Imputation of Cells (MAGIC) is an algorithm for denoising and transcript recover of single cells applied to single-cell RNA sequencing data, as described in van Dijk et al, 2018.

Usage

```
magic(data, ...)

## Default S3 method:
magic(data, genes = NULL, knn = 5, knn.max = NULL,
      decay = 1, t = 3, npca = 100, init = NULL, t.max = 20,
      knn.dist.method = "euclidean", verbose = 1, n.jobs = 1,
      seed = NULL, k = NULL, alpha = NULL, ...)

## S3 method for class 'seurat'
magic(data, genes = NULL, knn = 5, knn.max = NULL,
      decay = 1, t = 3, npca = 100, init = NULL, t.max = 20,
      knn.dist.method = "euclidean", verbose = 1, n.jobs = 1,
      seed = NULL, ...)

## S3 method for class 'Seurat'
magic(data, assay = NULL, genes = NULL, knn = 5,
      knn.max = NULL, decay = 1, t = 3, npca = 100, init = NULL,
      t.max = 20, knn.dist.method = "euclidean", verbose = 1,
      n.jobs = 1, seed = NULL, ...)
```

Arguments

data	input data matrix or Seurat object
...	Arguments passed to and from other methods
genes	character or integer vector, default: NULL vector of column names or column indices for which to return smoothed data. If 'all_genes' or NULL, the entire smoothed matrix is returned
knn	int, optional, default: 5 number of nearest neighbors on which to compute bandwidth
knn.max	int, optional, default: NULL maximum number of neighbors for each point. If NULL, defaults to 3*knn
decay	int, optional, default: 1 sets decay rate of kernel tails. If NULL, alpha decaying kernel is not used

t	int, optional, default: 3 power to which the diffusion operator is powered sets the level of diffusion. If 'auto', t is selected according to the Procrustes disparity of the diffused data.
npca	number of PCA components that should be used; default: 100.
init	magic object, optional object to use for initialization. Avoids recomputing intermediate steps if parameters are the same.
t.max	int, optional, default: 20 Maximum value of t to test for automatic t selection.
knn.dist.method	string, optional, default: 'euclidean'. recommended values: 'euclidean', 'cosine' Any metric from 'scipy.spatial.distance' can be used distance metric for building kNN graph.
verbose	'int' or 'boolean', optional (default : 1) If 'TRUE' or '> 0', print verbose updates.
n.jobs	'int', optional (default: 1) The number of jobs to use for the computation. If -1 all CPUs are used. If 1 is given, no parallel computing code is used at all, which is useful for debugging. For n_jobs below -1, (n.cpus + 1 + n.jobs) are used. Thus for n_jobs = -2, all CPUs but one are used
seed	int or 'NULL', random state (default: 'NULL')
k	Deprecated. Use 'knn'.
alpha	Deprecated. Use 'decay'.
assay	Assay to use for imputation, defaults to the default assay

Value

If a Seurat object is passed, a Seurat object is returned. Otherwise, a "magic" object containing: ****result****: matrix containing smoothed expression values ****operator****: The MAGIC operator (python magic.MAGIC object) ****params****: Parameters passed to magic

Examples

```
if (pymagic_is_available()) {
  data(magic_testdata)

  # Run MAGIC
  data_magic <- magic(magic_testdata, genes=c("VIM", "CDH1", "ZEB1"))
  summary(data_magic)
  ##           CDH1           VIM           ZEB1
  ## Min.      :0.4303   Min.      :3.854   Min.      :0.01111
  ## 1st Qu.:0.4444   1st Qu.:3.947   1st Qu.:0.01145
  ## Median :0.4462   Median :3.964   Median :0.01153
  ## Mean     :0.4461   Mean     :3.965   Mean     :0.01152
  ## 3rd Qu.:0.4478   3rd Qu.:3.982   3rd Qu.:0.01160
  ## Max.     :0.4585   Max.      :4.127   Max.     :0.01201

  # Plot the result with ggplot2
  if (require(ggplot2)) {
```

```
    ggplot(data_magic) +
      geom_point(aes(x=VIM, y=CDH1, color=ZEB1))
  }

  # Run MAGIC again returning all genes
  # We use the last run as initialization
  data_magic <- magic(magic_testdata, genes="all_genes", init=data_magic)
  # Extract the smoothed data matrix to use in downstream analysis
  data_smooth <- as.matrix(data_magic)

}

if (pymagic_is_available() && require(Seurat)) {

  data(magic_testdata)

  # Create a Seurat object
  seurat_object <- CreateSeuratObject(counts = t(magic_testdata), assay="RNA")
  seurat_object <- NormalizeData(object = seurat_object)
  seurat_object <- ScaleData(object = seurat_object)

  # Run MAGIC and reset the active assay
  seurat_object <- magic(seurat_object)
  seurat_object@active.assay = "MAGIC_RNA"

  # Analyze with Seurat
  VlnPlot(seurat_object, features=c("VIM", "ZEB1", "CDH1"))

}
```

magic_testdata *Fake scRNAseq data for examples*

Description

A subsampled dataset of epithelial to mesenchymal transition

Usage

```
magic_testdata
```

Format

A matrix with 500 rows and 197 variables

Source

The authors

print.magic *Print a MAGIC object*

Description

This avoids spamming the user's console with a list of many large matrices

Usage

```
## S3 method for class 'magic'  
print(x, ...)
```

Arguments

x	A fitted MAGIC object
...	Arguments for print()

Examples

```
if (pymagic_is_available()) {  
  
  data(magic_testdata)  
  data_magic <- magic(magic_testdata)  
  print(data_magic)  
  ## MAGIC with elements  
  ## $result : (500, 197)  
  ## $operator : Python MAGIC operator  
  ## $params : list with elements (data, knn, decay, t, npca, knn.dist.method)  
  
}
```

pymagic_is_available *Check whether MAGIC Python package is available and can be loaded*

Description

This is used primarily to avoid running tests on CRAN and elsewhere where the Python package should not be installed.

Usage

```
pymagic_is_available()
```

summary.magic	<i>Summarize a MAGIC object</i>
---------------	---------------------------------

Description

Summarize a MAGIC object

Usage

```
## S3 method for class 'magic'  
summary(object, ...)
```

Arguments

object	A fitted MAGIC object
...	Arguments for summary()

Examples

```
if (pymagic_is_available()) {  
  
  data(magic_testdata)  
  data_magic <- magic(magic_testdata)  
  summary(data_magic)  
  ## ZEB1  
  ## Min.   :0.01071  
  ## 1st Qu.:0.01119  
  ## Median :0.01130  
  ## Mean   :0.01129  
  ## 3rd Qu.:0.01140  
  ## Max.   :0.01201  
  
}
```

Index

*Topic **datasets**

magic_testdata, 7

as.data.frame.magic, 2

as.matrix.magic, 2

ggplot.magic, 3

install.magic, 3

library.size.normalize, 4

magic, 5

magic_testdata, 7

print.magic, 8

pymagic_is_available, 8

summary.magic, 9