Package ‘CellMapper’

February 1, 2026

Type Package

Title Predict genes expressed selectively in specific cell types

Version 1.37.0

Date 2016-10-05

Author Brad Nelms

Maintainer Brad Nelms <bnelms.research@gmail.com>

Description Infers cell type-specific expression based on co-expression similarity with known cell type marker genes. Can make accurate predictions using publicly available expression data, even when a cell type has not been isolated before.

License Artistic-2.0

Depends S4Vectors, methods

Suggests CellMapperData, Biobase, HumanAffyData, ALL, BiocStyle, ExperimentHub

Imports stats, utils

biocViews Microarray, Software, GeneExpression

git_url https://git.bioconductor.org/packages/CellMapper

git_branch devel

git_last_commit 1372904

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-02-01

Contents

`CellMapperList-class` *Class to contain expression data pre-processed for the 'CellMapper' package. The 'CellMapperList' class extends the 'SimpleList' class.*

Description

Container for expression data that has been pre-processed using the 'CMprep' function. A 'CellMapperList' object can be provided directly to the 'CMsearch' function to predict genes expressed selectively in specific cell types.

Details

'CMprep' transforms an expression matrix using singular value decomposition (SVD), resulting in a matrix, 'B', with the left-singular vectors of original data matrix and a vector, 'd', with the singular values. Singular vectors that account for less variance than an individual sample in the original dataset have been trimed (Kaiser's criterion), thereby removing singular vectors that mainly account for noise. The advantage of this transformation is that it reduces dataset size, and avoids the need to perform a time-consuming SVD transformation before running 'CMsearch'.

A 'CellMapperList' object contains the transformed left singular value matrix, 'B', and singular values, 'd', as well as meta-data associated with the original expression matrix.

Constructor:

'CellMapperList' instances are usually created through the 'CMprep' function. See ?CMprep

To create a 'CellMapperList' object directly, the following constructor is provided:

CellMapperList(B, d, meta = list())

where B is a numeric matrix containing left-singular vectors, d is a numeric vector containing singular values, and meta is a list object containing metadata.

Accessors

Same as for SimpleList objects. See ?SimpleList

The sample metadata can be extracted using metadata().

Author(s)

Brad Nelms

References

B.D. Nelms, L. Waldron, L.A. Barrera, A.W. Weflen, J.A. Goettel, G. Guo, R.K. Montgomery, M.R. Neutra, D.T. Breault, S.B. Snapper, S.H. Orkin, M.L. Bulyk, C. Huttenhower and W.I. Lencer. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome Biology 2016, 17(1).

See Also

CMsearch, CMprep

Examples

```
# Create a mock expression dataset with random expression values
ngenes <- 1000
narrays <- 100
x <- matrix(rnorm(ngenes*narrays), ngenes, narrays)
rownames(x) <- 1:ngenes

# Prepare a CellMapperList object using the CMprep function
data <- CMprep(x, DataSource = "Mock Expression Matrix")
show(data)
metadata(data)
```

CMprep *Pre-process a gene expression dataset for CellMapper*

Description

Prepares a dataset for use with the `CMsearch` function.

Usage

```
CMprep(Data, DataSource = '', GeneIDType = '', verbose = TRUE)
```

Arguments

`Data`	a numeric `matrix` or `ExpressionSet` object containing microarray expression data. The input expression data should be normalized and log-transformed.
`DataSource`	An optional character vector of length one providing some information about the expression data source.
`GeneIDType`	An optional character vector of length one that lists the type of gene IDs used in the expression dataset (e.g. "Human Entrez IDs").
`verbose`	logical value indicating whether progress updates should be provided.

Details

This function calculates the singular value decomposition of a gene expression matrix and prepares the data for use with the `CMsearch` function. It can take some time for large expression matrices, and so it is recommended to save the output for future use. Pre-processed microarray data, ready for immediate use with `CMsearch`, can be found in the `CellMapperData` package.

See the CellMapper vignette for examples about how to use `CMprep` and associated functions to infer genes selectively expressed in specific cell types.

Value

A 'CellMapperList' object ready to be provided to the `CMsearch` function.

Author(s)

Brad Nelms

References

B.D. Nelms, L. Waldron, L.A. Barrera, A.W. Weflen, J.A. Goettel, G. Guo, R.K. Montgomery, M.R. Neutra, D.T. Breault, S.B. Snapper, S.H. Orkin, M.L. Bulyk, C. Huttenhower and W.I. Lencer. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome Biology 2016, 17(1).

See Also

`CMsearch`, `ExpressionSet`, `CellMapperData`

Examples

```
# Create a mock expression dataset with random expression values
ngenes <- 1000
narrays <- 100
x <- matrix(rnorm(ngenes*narrays), ngenes, narrays)
rownames(x) <- 1:ngenes

# Prepare the dataset for use with CMsearch
data <- CMprep(x)
show(data)

# Save the processed dataset for later
## Not run:
save(data, file = "Preprocessed_CellMapper_Data.RData")

## End(Not run)
```

`CMsearch` *Predict cell type-expressed genes using CellMapper*

Description

Predicts which genes are selectively expressed in the same cell type as a given cell type-specific marker gene (the 'query gene'), based on co-expression similarity.

Usage

```
CMsearch(Data, query.genes = NULL, control.genes = NULL, QDW = TRUE,
alpha = 0.5, verbose = TRUE, raw.pvals = FALSE)
```

Arguments

`Data`	a `CellMapperList` object containing expression data processed with `CMprep`, or a list of datasets processed with `CMprep`.
`query.genes`	a list of genes that are specifically expressed in the cell type of interested, supplied as a character vector of gene names (matching the row names of the original expression matrix).
`control.genes`	a list of genes expressed specifically in non-target cell types (optional), supplied as a character vector of gene names (matching the row names of the original expression matrix). This option generally has little effect on the results and its use is NOT recommended.
`QDW`	logical value indicating whether 'query driven weighting' should be applied in the CellMapper SVD filter. The default value of TRUE can be used in most cases.
`alpha`	alpha parameter controlling the strength of the CellMapper SVD filter. The default value of 0.5 can be used in most cases.
`verbose`	logical value indicating whether progress updates should be provided.
`raw.pvals`	logical value indicating whether unadjusted p-values, which have not been corrected for multiple hypothesis testing, should be returned.

Details

This function predicts which genes are selectively expressed in the same cell type as a given cell type-specific marker gene (the 'query gene'), based on co-expression similarity. The only required inputs are a gene expression matrix that has been pre-processed with the `CMprep` function (or a list of pre-processed expression matrices), and a 'query gene' known to be specifically expressed in the cell type of interest. Pre-processed microarray data, ready for immediate use with `CMsearch`, can be found in the `CellMapperData` package.

See the CellMapper vignette for examples about how to use `CMsearch` and associated functions to infer genes selectively expressed in specific cell types.

Value

A dataframe containing the gene identifiers in the first column, the false discovery rate (FDR) in the second, and the unadjusted p-value in the third. FDR is calculated using the method of Benjamini and Hochberg.

Author(s)

Brad Nelms

References

B.D. Nelms, L. Waldron, L.A. Barrera, A.W. Weflen, J.A. Goettel, G. Guo, R.K. Montgomery, M.R. Neutra, D.T. Breault, S.B. Snapper, S.H. Orkin, M.L. Bulyk, C. Huttenhower and W.I. Lencer. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome Biology 2016, 17(1).

See Also

CMprep, CellMapperData

Examples

```
# Create a mock expression dataset with random expression values
ngenes <- 1000
narrays <- 100
x <- matrix(rnorm(ngenes*narrays), ngenes, narrays)
rownames(x) <- 1:ngenes

# Prepare the dataset for use with CMsearch
data <- CMprep(x)

# Predict which genes are co-expressed in the same cell type as 'gene' 59 in the
# mock expression dataset
results <- CMsearch(data, query.genes = '59')

head(results)
```

ExampleGeneLists *Example Gene Lists*

Description

Query genes, positive control genes, and negative control genes for example cell types.

Usage

```
data(ExampleQueryGenes)
data(PositiveControlGenes)
data(NegativeControlGenes)
```

Details

'ExampleQueryGenes' is a data frame listing the cell types and query genes analyzed in Nelms, et al. (2016). These genes can be used as query or control genes for custom CellMapper searches, and are also provided for reproducibility of the manuscript results. All predictions from the manuscript can be generated using the `ReplicateManuscript` function. The 'Dataset' column indicates which dataset(s) were used for each cell type, available from Bioconductor's `ExperimentHub`. All datasets are available from the `CellMapperData` package.

'PositiveControlGenes' is a list of positive control genes for each cell type used in Nelms, et al. (2016). These positive control genes were NOT provided to the algorithm, but were used to assess the accuracy of the results. This list contains Entrez gene IDs.

'NegativeControlGenes' is a list of negative control genes for each cell type used in Nelms, et al. (2016). These negative control genes were NOT provided to the algorithm, but were used to assess the accuracy of the results. This list contains Entrez gene IDs.

Value

'ExampleQueryGenes' is a data frame, 'PositiveControlGenes' and 'NegativeControlGenes' are both lists of character vectors

Author(s)

Brad Nelms

References

B.D. Nelms, L. Waldron, L.A. Barrera, A.W. Weflen, J.A. Goettel, G. Guo, R.K. Montgomery, M.R. Neutra, D.T. Breault, S.B. Snapper, S.H. Orkin, M.L. Bulyk, C. Huttenhower and W.I. Lencer. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome Biology 2016, 17(1).

See Also

`CMsearch`, `ReplicateManuscript`, `CellMapperData`, `ExperimentHub`

Examples

```
data(ExampleQueryGenes)
head(ExampleQueryGenes)
```

`ReplicateManuscript` *Replicate Results from CellMapper Manuscript*

Description

Recreates the CellMapper predictions from Nelms, et al. (2016) – Additional dataset 1.

Usage

```
ReplicateManuscript()
```

Details

This function will recreate the CellMapper predictions from Nelms, et al. (2016), Additional dataset 1. It requires the `ExperimentHub` and `CellMapperData` packages to be installed. Provided that the `CellMapperData` resources have previously been downloaded from `ExperimentHub`, the 'ReplicateManuscript' function should take ~5 minutes to run on a personal laptop.

Value

A 'list' object containing CellMapper predictions for each cell type. Genes names are returned as Entrez identifiers.

Author(s)

Brad Nelms

References

B.D. Nelms, L. Waldron, L.A. Barrera, A.W. Weflen, J.A. Goettel, G. Guo, R.K. Montgomery, M.R. Neutra, D.T. Breault, S.B. Snapper, S.H. Orkin, M.L. Bulyk, C. Huttenhower and W.I. Lencer. CellMapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. Genome Biology 2016, 17(1).

See Also

`CMsearch`, `ExampleQueryGenes`, `CellMapperData`, `ExperimentHub`

Examples

```
Results <- ReplicateManuscript()
head(Results[['GABAergic Neurons']])
```

Index