Package 'scry'

October 17, 2020

Title Small-Count Analysis Methods for High-Dimensional Data

Version 1.0.1

Description Many modern biological datasets consist of small counts that are not well fit by standard linear-Gaussian methods such as principal component analysis. This package provides implementations of count-based feature selection and dimension reduction algorithms. These methods can be used to facilitate unsupervised analysis of any high-dimensional data such as single-cell RNA-seq.

License Artistic-2.0

Depends R (>= 4.0), stats, methods

Imports glmpca (>= 0.2.0), Matrix, SingleCellExperiment, SummarizedExperiment

Suggests BiocGenerics, knitr, testthat, covr, rmarkdown, ggplot2, DuoClustering2018

VignetteBuilder knitr

LazyData false

URL https://bioconductor.org/packages/scry.html

BugReports https://github.com/kstreet13/scry/issues

RoxygenNote 7.1.1

Encoding UTF-8

biocViews DimensionReduction, GeneExpression, Normalization, PrincipalComponent, RNASeq, Software, Sequencing, SingleCell, Transcriptomics

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

git_branch RELEASE_3_11

git_last_commit f2cc014

git_last_commit_date 2020-07-28

Date/Publication 2020-10-16

Author F. William Townes [aut, cre, cph], Kelly Street [aut]

Maintainer F. William Townes <will.townes@gmail.com>

R topics documented:

`compute_size_factors` *Compute size factors*

Description

Computes a size factor for each observation (column) of a count data matrix based on an approximate multinomial model.

Usage

```
compute_size_factors(m, fam = c("binomial", "poisson"))
```

Arguments

`m`	a matrix or sparse `Matrix` of integer count values.
`fam`	a string specifying the model type to be used for calculating size factors. Must be either 'binomial' or 'poisson'.

Details

Both fam options are approximations to a multinomial model. Size factors for binomial are simply the column sums (total counts for each sample). For Poisson, the size factors are given by the column sums after rescaling to have geometric mean of one. This improves numerical stability.

Value

A vector of size factors with length equal to the number of columns of `m`.

Examples

```
ncells <- 100
u <- matrix(rpois(20000, 5), ncol=ncells)
compute_size_factors(u)
```

`devianceFeatureSelection` *Feature selection by approximate multinomial deviance*

Description

Computes a deviance statistic for each row feature (such as a gene) for count data based on a multinomial null model that assumes each feature has a constant rate. Features with large deviance are likely to be informative. Uninformative, low deviance features can be discarded to speed up downstream analyses and reduce memory footprint.

Usage

```
devianceFeatureSelection(object, ...)

## S4 method for signature 'SummarizedExperiment'
devianceFeatureSelection(
  object,
  assay = 1,
  fam = c("binomial", "poisson"),
  batch = NULL,
  nkeep = NULL,
  sorted = FALSE
)

## S4 method for signature 'matrix'
devianceFeatureSelection(object, fam = c("binomial", "poisson"), batch = NULL)

## S4 method for signature 'Matrix'
devianceFeatureSelection(object, fam = c("binomial", "poisson"), batch = NULL)
```

Arguments

`object`	an object inheriting from SummarizedExperiment (such as `SingleCellExperiment`). Alternatively, a matrix or sparse Matrix of integer counts.
`...`	for the generic, additional arguments to pass to object-specific methods.
`assay`	a string or integer specifying which assay contains the count data (default = 1). Ignored if `object` is a matrix or sparse Matrix.
`fam`	a string specifying the model type to be used for calculating the residuals. Binomial (the default) is the closest approximation to multinomial, but Poisson may be faster to compute and often is very similar to binomial.
`batch`	an optional factor indicating batch membership of observations. If provided, the null model is computed within each batch separately to regress out the batch effect from the resulting deviance statistics.
`nkeep`	integer, how many informative features should be retained? Default: all features are retained if set to NULL. Ignored if `object` is a matrix or sparse Matrix.
`sorted`	logical, should the `object` be returned with rows sorted in decreasing order of deviance? Default: FALSE, unless nkeep is specified, in which case it is forced to be TRUE. Ignored for matrix and sparse Matrix inputs.

Details

In a typical single-cell analysis, many of the features (genes) may not be informative about differences between observations (cells). Feature selection seeks to identify which genes are the most informative. We define an informative gene as one that is poorly fit by a multinomial model of constant expression across cells within each batch. We compute a deviance statistic for each gene. Genes with high deviance are more informative.

Value

The original `SingleCellExperiment` or `SummarizedExperiment` object with the deviance statistics for each feature appended to the rowData. The new column name will be either binomial_deviance or poisson_deviance. If the input was a matrix or sparse Matrix, output is a numeric vector containing the deviance statistics for each row.

References

Townes FW, Hicks SC, Aryee MJ, and Irizarry RA (2019). Feature Selection and Dimension Reduction for Single Cell RNA-Seq based on a Multinomial Model. *Genome Biology* https://doi.org/10.1186/s13059-019-1861-6

Examples

```
ncells <- 100
u <- matrix(rpois(20000, 5), ncol=ncells)
sce <- SingleCellExperiment::SingleCellExperiment(assays=list(counts=u))
devianceFeatureSelection(sce)
```

`GLMPCA` *Generalized principal components analysis for non-normally distributed data*

Description

This function implements the GLM-PCA dimensionality reduction method for high-dimensional count data. This is a wrapper for `glmpca`.

Usage

```
GLMPCA(object, ...)

## S4 method for signature 'SummarizedExperiment'
GLMPCA(object, assay = 1, L, ...)

## S4 method for signature 'matrix'
GLMPCA(object, assay = 1, L, ...)

## S4 method for signature 'Matrix'
GLMPCA(object, assay = 1, L, ...)
```

Arguments

object	A SingleCellExperiment or SummarizedExperiment object. Alternatively, a matrix of integer counts.
...	further arguments passed to glmpca
assay	a character or integer specifying which assay to use for GLM-PCA (default = 1). Ignored if object is a matrix.
L	the desired number of latent dimensions (integer).

Value

The original SingleCellExperiment or SummarizedExperiment object with the GLM-PCA results added to the metadata slot. If the original input was a SingleCellExperiment, then a new reducedDim element called "GLMPCA" will be added, representing the GLM-PCA factors. If the input was a matrix, output matches that of glmpca.

Examples

```
ncells <- 100
u <- matrix(rpois(20000, 5), ncol=ncells)
sce <- SingleCellExperiment::SingleCellExperiment(assays=list(counts=u))
GLMPCA(sce, L = 2)
```

---

nullResiduals *Residuals from an approximate multinomial null model*

Description

Computes deviance or Pearson residuals for count data based on a multinomial null model that assumes each feature has a constant rate. The residuals matrix can be analyzed with standard PCA as a fast approximation to GLM-PCA.

Usage

```
nullResiduals(object, ...)

## S4 method for signature 'SummarizedExperiment'
nullResiduals(
  object,
  assay = 1,
  fam = c("binomial", "poisson"),
  type = c("deviance", "pearson"),
  batch = NULL
)

## S4 method for signature 'matrix'
nullResiduals(
  object,
  fam = c("binomial", "poisson"),
  type = c("deviance", "pearson"),
```

```
  batch = NULL
)

## S4 method for signature 'Matrix'
nullResiduals(
  object,
  fam = c("binomial", "poisson"),
  type = c("deviance", "pearson"),
  batch = NULL
)
```

Arguments

object	an object inheriting from SummarizedExperiment (such as SingleCellExperiment). Alternatively, a matrix of integer counts.
...	for the generic, additional arguments to pass to object-specific methods.
assay	a string or integer specifying which assay contains the count data (default = 1). Ignored if object is a matrix.
fam	a string specifying the model type to be used for calculating the residuals. Binomial (the default) is the closest approximation to multinomial, but Poisson may be faster to compute and often is very similar to binomial.
type	should deviance or Pearson residuals be used?
batch	an optional factor indicating batch membership of observations. If provided, the null model is computed within each batch separately to regress out the batch effect from the resulting residuals.

Details

This function should be used only on the un-normalized counts. It was originally designed for single-cell RNA-seq counts obtained by the use of unique molecular identifiers (UMIs) and has not been tested on read count data without UMIs or other data types.

Note that even though sparse Matrix objects are accepted as input, they are internally coerced to dense matrix before processing, because the output is always a dense matrix since the residuals transformation is not sparsity preserving. To avoid memory issues, it is recommended to perform feature selection first and subset the number of features to a smaller size prior to computing the residuals.

Value

The original SingleCellExperiment or SummarizedExperiment object with the residuals appended as a new assay. The assay name will be fam_type_residuals (eg, binomial_deviance_residuals). If the input was a matrix, output is a dense matrix containing the residuals.

References

Townes FW, Hicks SC, Aryee MJ, and Irizarry RA (2019). Feature Selection and Dimension Reduction for Single Cell RNA-Seq based on a Multinomial Model. *Genome Biology* https://doi.org/10.1186/s13059-019-1861-6

Examples

```
ncells <- 100
u <- matrix(rpois(20000, 5), ncol=ncells)
sce <- SingleCellExperiment::SingleCellExperiment(assays=list(counts=u))
nullResiduals(sce)
```

Index