Package ‘slalom’

October 14, 2021

Type Package

Title Factorial Latent Variable Modeling of Single-Cell RNA-Seq Data

Version 1.14.0

Date 2018-04-25

Description slalom is a scalable modelling framework for single-cell RNA-seq data that uses gene set annotations to dissect single-cell transcriptome heterogeneity, thereby allowing to identify biological drivers of cell-to-cell variability and model confounding factors.

Depends R (>= 3.4)

Imports Rcpp (>= 0.12.8), RcppArmadillo, BH, ggplot2, grid, GSEABase, methods, rsvd, SingleCellExperiment, SummarizedExperiment, stats

Suggests knitr, rhdf5, scater, testthat

LinkingTo Rcpp, RcppArmadillo, BH

License GPL-2

VignetteBuilder knitr

LazyData true

Encoding UTF-8

biocViews ImmunoOncology, SingleCell, RNASeq, Normalization, Visualization, DimensionReduction, Transcriptomics, GeneExpression, Sequencing, Software, Reactome

RoxygenNote 6.0.1

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

git_branch RELEASE_3_13

git_last_commit 926621e

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

Author Florian Buettner [aut],
Naruemon Pratanwanich [aut],
Davis McCarthy [aut, cre],

John Marioni [aut],
Oliver Stegle [aut]

Maintainer Davis McCarthy <davis@ebi.ac.uk>

R topics documented:

addResultsToSingleCellExperiment

Add results to SingleCellExperiment object

Description

Add results to SingleCellExperiment object

Usage

```
addResultsToSingleCellExperiment(sce_object, slalom_object, n_active = 20,
  mad_filter = 0.4, annotated = TRUE, unannotated_dense = FALSE,
  unannotated_sparse = FALSE, add_loadings = TRUE, dimred = "slalom",
  check_convergence = TRUE)
```

Arguments

sce_object	an object of class SingleCellExperiment
slalom_object	an object of class Rcpp_SlalomModel
n_active	number of terms (factors) to be added (default is 20)
mad_filter	numeric(1), filter factors by this mean absolute deviation to ensure variability in the factor states. For large datasets this can be set to 0
annotated	logical(1), should annotated factors be included? Default is TRUE

`unannotated_dense`
: logical(1), should dense unannotated factors be included? Default is `FALSE`

`unannotated_sparse`
: logical(1), should sparse unannotated factors be included? Default is `FALSE`

`add_loadings`
: logical(1), should gene/feature loadings be added to the `rowData` of the object?

`dimred`
: character(1), name of the reduced-dimension slot to save the factor states to. Default is `"slalom"`

`check_convergence`
: logical(1), check that model has converged before adding `slalom` results. If `TRUE` and model has not converged it throws an error.

Value

a `SingleCellExperiment` object with factor states (X) in a reduced-dimension slot, and gene loadings for factors added to `rowData`.

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
mesc <- addResultsToSingleCellExperiment(mesc, model,
check_convergence = FALSE)
```

`initSlalom` *Initialize a SlalomModel object*

Description

Initialize a SlalomModel with sensible starting values for parameters before training the model.

Usage

```
initSlalom(object, alpha_priors = NULL, epsilon_priors = NULL,
  noise_model = "gauss", seed = NULL, pi_prior = NULL, n_hidden = NULL,
  design = NULL, verbose = FALSE, save_init = FALSE)
```

Arguments

`object`
: a `Rcpp_SlalomModel` object

`alpha_priors`
: numeric(2) giving alpha and beta hyperparameters for a gamma prior distribution for alpha parameters (precision of factor weights)

`epsilon_priors`
: numeric(2) giving alpha and beta hyperparameters for a gamma prior distribution for noise precision parameters

`noise_model` character(1) defining noise model, defaults to "gauss" for Gaussian noise model

`seed` integer(1) value supplying a random seed to make results reproducible (default is `NULL`)

`pi_prior` numeric matrix (genes x factors) giving prior probability of a gene being active for a factor

`n_hidden` integer(1), number of hidden factors in model. Required if `pi_prior` is not `NULL`, ignored otherwise.

`design` matrix of known factors (covariates) to fit in the model. Optional if `pi_prior` is not `NULL`, ignored otherwise.

`verbose` logical(1), should messages be printed about what the function is doing? Default is `TRUE`.

`save_init` logical(1), save the initial X values (factor states for each cell) in the object? Default is `FALSE` as this is potentially a large N (number of cell) by K (number of factors) matrix.

Details

It is strongly recommended to use `newSlalomModel` to create the `SlalomModel` object prior to applying `initSlalom`.

Value

an ‘Rcpp_SlalomModel‘ object

Author(s)

Davis McCarthy

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
```

`mesc` *A single-cell expression dataset to demonstrate capabilities of slalom from mouse embryonic stem cells (mESCs)*

Description

This data set consists of an `SCESet` object with log2-counts-per-million expression values for 3635 genes for 182 cells. They are from a real experiment, studying cell cycle in mouse embryonic stem cells (mESCs). See Buettner et al (Nat. Biotech., 2015) for details. d.

Usage

```
mesc
```

Format

an SCESet instance, 1 row per gene.

Value

NULL, but makes aavailable an SCESet object containing expression data

Author(s)

Davis McCarthy, Florian Buettner, 2016-12-02

Source

EMBL-EBI, Hinxton, UK

References

Buettner F, Natarajan KN, Paolo Casale F, Proserpio V, Scialdone A, Theis FJ, et al. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. Nat Biotechnol. Nature Publishing Group; 2015;33: 155–160.

`newSlalomModel` *Create a new SlalomModel object.*

Description

Slalom fits relatively complicated hierarchical Bayesian factor analysis models with data and results stored in a `"SlalomModel"` object. This function builds a new `"SlalomModel"` object from minimal inputs.

Usage

```
newSlalomModel(object, genesets, n_hidden = 5, prune_genes = TRUE,
  min_genes = 15, design = NULL, anno_fpr = 0.01, anno_fnr = 0.001,
  assay_name = "logcounts", verbose = TRUE)
```

Arguments

`object`	`"SingleCellExperiment"` object N x G expression data matrix (cells x genes)
`genesets`	a `"GeneSetCollection"` object containing annotated gene sets
`n_hidden`	number of hidden factors to fit in the model (2-5 recommended)
`prune_genes`	logical, should genes that are not annotated to any gene sets be filtered out? If `TRUE`, then any genes with zero variance in expression are also filtered out.

`min_genes`	scalar, minimum number of genes required in order to retain a gene set for analysis
`design`	numeric design matrix providing values for covariates to fit in the model (rows represent cells)
`anno_fpr`	numeric(1), false positive rate (FPR) for assigning genes to factors (pathways); default is 0.01
`anno_fnr`	numeric(1), false negative rate (FNR) for assigning genes to factors (pathways); default is 0.001
`assay_name`	character(1), the name of the `assay` of the `object` to use as expression values. Default is `logcounts`, assumed to be normalised log2-counts-per-million values or equivalent.
`verbose`	logical(1), should information about what's going be printed to screen?

Details

This function builds and returns the object, checking for validity, which includes checking that the input data is of consistent dimensions.

Value

a new Rcpp_SlalomModel object

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)

exprsfile <- system.file("extdata", "mesc.csv", package = "slalom")
mesc_mat <- as.matrix(read.csv(exprsfile))
sce <- SingleCellExperiment::SingleCellExperiment(assays = list(logcounts = mesc_mat))
# model2 <- newSlalomModel(mesc_mat, genesets, n_hidden = 5, min_genes = 10)
```

`plotLoadings` *Plot highest loadings of a factor*

Description

Plot highest loadings of a factor

Usage

```
plotLoadings(object, term, n_genes = 10)
```

Arguments

object	an object of class Rcpp_SlalomModel
term	integer(1) or character(1), providing either index for desired term (if an integer) or the term name (if character)
n_genes	integer(1), number of loadings (genes) to show

Details

Show the factor loadings for a genes with the highest loadings for a given factor. Absolute weights are shown, with genes ordered by absolute weight. Indications are given on the plot as to whether the gene was originally in the factor geneset or added to it by the slalom model.

Value

a ggplot plot object

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
plotLoadings(model, term = 2)
```

plotRelevance *Plot results of a Slalom model*

Description

Plot results of a Slalom model

Usage

```
plotRelevance(object, n_active = 20, mad_filter = 0.4, annotated = TRUE,
  unannotated_dense = FALSE, unannotated_sparse = FALSE)
```

Arguments

object	an object of class Rcpp_SlalomModel
n_active	number of terms (factors) to be plotted (default is 20)
mad_filter	numeric(1), filter factors by this mean absolute deviation to exclude outliers. For large datasets this can be set to 0
annotated	logical(1), should annotated factors be plotted? Default is TRUE

`unannotated_dense`
: logical(1), should dense unannotated factors be plotted? Default is `FALSE`

`unannotated_sparse`
: logical(1), should sparse unannotated factors be plotted? Default is `FALSE`

Value

invisibly returns a list containing the two ggplot objects that make up the plot

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
plotRelevance(model)
```

`plotTerms` *Plot relevance for all terms*

Description

Plot relevance for all terms

Usage

```
plotTerms(object, terms = NULL, order_terms = TRUE, mad_filter = 0.2,
  annotated = TRUE, unannotated_dense = TRUE, unannotated_sparse = FALSE)
```

Arguments

`object`
: an object of class `Rcpp_SlalomModel`

`terms`
: integer or character vector, providing either indices for desired terms (if an integer) or the term names (if character); default is `NULL`, in which case all terms are plotted.

`order_terms`
: logical(1), should factors be ordered by relevance (`TRUE`, default), or in the order the come

`mad_filter`
: numeric(1), filter factors by this mean absolute deviation to exclude outliers. For large datasets this can be set close to 0; default is `0.2`.

`annotated`
: logical(1), should annotated factors be plotted? Default is `TRUE`

`unannotated_dense`
: logical(1), should dense unannotated factors be plotted? Default is `TRUE`

`unannotated_sparse`
: logical(1), should sparse unannotated factors be plotted? Default is `TRUE`

Value

a ggplot plot object

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
plotTerms(model)
```

`Rcpp_SlalomModel` *The "Slalom Model" (Rcpp_SlalomModel) class*

Description

S4 class and the main class used by slalom to hold model data and results. SingleCellExperiment extends the Bioconductor SummarizedExperiment class.

Details

This class is initialized from a matrix of expression values and a collection of genesets in a `GeneSetCollection` object from the GSEABase package.

Methods that operate on SingleCellExperiment objects constitute the basic scater workflow.

Methods

`train()` void train() docstring : Train the SlalomModel

`update()` void update() docstring : Update the SlalomModel

`updateAlpha(...)` void updateAlpha(int) docstring : Update alpha

`updateEpsilon()` void updateEpsilon() docstring : Update Epsilon

`updatePi(...)` void updatePi(int) docstring : Update Pi

`updateW(...)` void updateW(int) docstring : Update W

`updateX(...)` void updateX(int) docstring : Update X

Slots

`.xData`**:** Environment enabling access to the C++-level SlalomModel object.

`slalom` *Factorial single-cell latent variable models*

Description

slalom

Details

Factorial latent variable models for RNA-seq data.

Author(s)

Davis McCarthy

`SlalomModel` *SlalomModel C++ class*

Description

A C++ class for SlalomModel models.

Arguments

`Y_init`	matrix of expression values
`pi_init`	G x K matrix with each entry being the prior probability for a gene g being active for factor k.
`X_init`	matrix of initial factor states (N x K)
`W_init`	G x K matrix of initial weights
`prior_alpha`	numeric vector of length two giving prior values for the gamma hyperparameters of the precisions
`prior_epsilon`	numeric vector of length two giving prior values for the gamma hyperparameters of the residual variances

Value

an object of the SlalomModel class

topTerms *Show results of a Slalom model*

Description

Show results of a Slalom model

Usage

```
topTerms(object, n_active = 20, mad_filter = 0.4, annotated = TRUE,
  unannotated_dense = FALSE, unannotated_sparse = FALSE)
```

Arguments

`object`	an object of class `Rcpp_SlalomModel`
`n_active`	number of terms (factors) to be plotted (default is 20)
`mad_filter`	numeric(1), filter factors by this mean absolute deviation to exclude outliers. For large datasets this can be set to 0
`annotated`	logical(1), should annotated factors be plotted? Default is `TRUE`
`unannotated_dense`	logical(1), should dense unannotated factors be plotted? Default is `FALSE`
`unannotated_sparse`	logical(1), should sparse unannotated factors be plotted? Default is `FALSE`

Value

data.frame with factors ordered by relevance, showing `term` (term names), `relevance`, `type` (factor type: known, annotated or unannotated), `n_prior` (number of genes annotated to the gene set/factor), `n_gain` (number of genes added/switched on for the factor), `n_loss` (number of genes turned off for the factor).

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
topTerms(model)
```

trainSlalom *Train a SlalomModel object*

Description

Train a SlalomModel to infer model parameters.

Usage

```
trainSlalom(object, nIterations = 5000, minIterations = 700,
  tolerance = 1e-08, forceIterations = FALSE, shuffle = TRUE,
  pretrain = TRUE, verbose = TRUE, seed = NULL, drop_factors = TRUE)
```

Arguments

`object`	a `Rcpp_SlalomModel` object
`nIterations`	integer(1) maximum number of iterations to use in training the model (default: 5000)
`minIterations`	integer(1) minimum number of iterations to perform.
`tolerance`	numeric(1) tolerance to allow between iterations (default 1e-08)
`forceIterations`	logical(1) should the model be forced to update `nIteration` times?
`shuffle`	logical(1) should the order in which factors are updated be shuffled between iterations? Shuffling generally helps speed up convergence so is recommended and defaults is `TRUE`
`pretrain`	logical(1), should the model be "pre-trained" to achieve faster convergence and obtain an initial update order? Recommended; default is `TRUE`
`verbose`	logical(1), should messages be printed about what the function is doing? Default is `TRUE`.
`seed`	integer(1) value supplying a random seed to make results reproducible (default is `NULL`)
`drop_factors`	logical(1), should factors be dropped from the model if the model determines them not to be relevant? Default is `TRUE`.

Details

Train the model using variational Bayes methods to infer parameters.

Value

an ‘Rcpp_SlalomModel‘ object

Author(s)

Davis McCarthy

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- trainSlalom(model, nIterations = 10)
```

---

updateSlalom *Update a SlalomModel object*

Description

Do one variational update of a SlalomModel to infer model parameters.

Usage

```
updateSlalom(object)
```

Arguments

object	a Rcpp_SlalomModel object

Details

Update the model with one iteration using variational Bayes methods to infer parameters.

Value

an ‘Rcpp_SlalomModel‘ object

Author(s)

Davis McCarthy

Examples

```
gmtfile <- system.file("extdata", "reactome_subset.gmt", package = "slalom")
genesets <- GSEABase::getGmt(gmtfile)
data("mesc")
model <- newSlalomModel(mesc, genesets, n_hidden = 5, min_genes = 10)
model <- initSlalom(model)
model <- updateSlalom(model)
```

Index