Package 'iSEEfier'

November 12, 2026

Title Streamlining the creation of initial states for starting an iSEE instance

Version 1.8.0

Description iSEEfier provides a set of functionality to quickly and intuitively create, inspect, and combine initial configuration objects.
These can be conveniently passed in a straightforward manner to the function call to launch iSEE() with the specified configuration.
This package currently works seamlessly with the sets of panels provided by the iSEE and iSEEu packages, but can be extended to accommodate the usage of any custom panel (e.g. from iSEEde, iSEEpathways, or any panel developed independently by the user).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports iSEE, iSEEu, methods, ggplot2, igraph, rlang, stats, SummarizedExperiment, SingleCellExperiment, visNetwork, BiocBaseUtils

Suggests knitr, rmarkdown, scater, scRNAseq, org.Mm.eg.db, scuttle, BiocStyle, testthat (>= 3.0.0)

biocViews CellBasedAssays, Clustering, DimensionReduction, FeatureExtraction, GUI, GeneExpression, ImmunoOncology, ShinyApps, SingleCell, Software, Transcription, Transcriptomics, Visualization

URL https://github.com/NajlaAbassi/iSEEfier

BugReports https://github.com/NajlaAbassi/iSEEfier/issues

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch RELEASE_3_23

git_last_commit 0eb4919

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Najla Abassi [aut, cre] (ORCID: <https://orcid.org/0000-0001-8357-0938>), Federico Marini [aut] (ORCID: <https://orcid.org/0000-0003-3252-7758>)

Maintainer Najla Abassi <abassi.nejla96@gmail.com>

Contents

constants-iSEEfier 2
glue_initials 3
iSEEfier-pkg 4
iSEEinit 5
iSEEmarker 6
iSEEnrich 7
view_initial_network 8
view_initial_tiles 9

Index **11**

`constants-iSEEfier` *Constant values used throughout iSEEfier*

Description

Constant values used throughout iSEEfier

Usage

```
iSEE_panel_colors
```

Format

An object of class `character` of length 17.

Panel colors

- color values (as string character or hex value) for the panels included by default in `iSEE` and `iSEEu`

glue_initials *Glue together initial objects into one*

Description

Glue a set of `initial` configuration objects, combining them into a single valid `initial` set.

Usage

```
glue_initials(
  ...,
  remove_duplicate_panels = TRUE,
  verbose = TRUE,
  custom_panels_allowed = NULL
)
```

Arguments

`...`	A set of `initial` list objects (in the format that is required to be passed as a parameter in the call to `iSEE::iSEE()`) - just as in the behavior of the `c()`/`paste()` function
`remove_duplicate_panels`	Logical, defaults to `TRUE`. Defines the behavior to remove panels detected as duplicated. Can be relevant upon concatenating mid to large sets of panels.
`verbose`	Logical, defaults to `TRUE`. If on, prints out a series of informative messages to describe the actions undertaken upon running.
`custom_panels_allowed`	Character vector, defaults to `NULL`. Can be used to specify additional panels to be allowed in the concatenation.

Details

The usage of `custom_panels_allowed` can be especially relevant when one creates one or more custom panels, with a specific name that needs to be indicated in this parameter. For example, if using a panel of class `FancyPlotPanel` and one called `FancyTablePanel`, the value for `custom_panels_allowed` should be set to `c("FancyPlotPanel", "FancyTablePanel")`.

It is worth mentioning that `iSEE::iSEE()` is actually able to handle the automatic renaming of panels that could be detected as duplicated. This can basically relax the requirement on the "uniqueness" of the configured panels, with the only caveat of having to think of how the *transmissions* between panels will be handled; nevertheless, most users might not even need to face this situation.

Value

A single `initial` list object, in the format that is required to be passed as a parameter in the call to `iSEE::iSEE()`, concatenating the values provided as input.

Examples

```
## Load a dataset and preprocess this quickly
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
sce <- scater::runTSNE(sce)
## Select some features and aspects to focus on
gene_list_1 <- c("ENSMUSG00000026581")
gene_list_2 <- c("ENSMUSG00000005087", "ENSMUSG00000015437")
cluster <- "stimulus"
group <- "single cell quality"
initial1 <- iSEEinit(sce = sce,
                     features = gene_list_1,
                     clusters = cluster,
                     groups = group)
initial2 <- iSEEinit(sce = sce,
                     features = gene_list_2,
                     clusters = cluster,
                     groups = group)
initials_merged <- glue_initials(initial1,
                                 initial2)
view_initial_tiles(initial1)
view_initial_tiles(initial2)
view_initial_tiles(initials_merged)

## Continue your exploration directly within iSEE!
if (interactive())
  iSEE(sce, initial = initial_merged)
```

`iSEEfier-pkg` *iSEEfier: a very convenient way to fire up your iSEE instance*

Description

iSEEfier provides a set of functionality to quickly create, inspect, and combine `initial` configuration objects. These can be conveniently passed to the function call to launch `iSEE()` in this manner. This currently works with the sets of panels provided by the `iSEE` and `iSEEu` packages, but can be extended to accommodate the usage of any custom panel (e.g. from `iSEEde`, `iSEEpathways`, or any panel developed independently by the user).

Author(s)

Maintainer: Najla Abassi <abassi.nejla96@gmail.com> (ORCID)

Authors:

- Federico Marini <marinif@uni-mainz.de> (ORCID)

See Also

Useful links:

- https://github.com/NajlaAbassi/iSEEfier
- Report bugs at https://github.com/NajlaAbassi/iSEEfier/issues

---

`iSEEinit` *iSEEinit: Create an initial state of an iSEE instance for gene expression visualization*

Description

`iSEEinit()` defines the initial setup of an iSEE instance, recommending tailored visual elements to effortlessly illustrate the expression of a gene list in a single view.

Usage

```
iSEEinit(
  sce,
  features,
  reddim_type = "TSNE",
  clusters = colnames(colData(sce))[1],
  groups = colnames(colData(sce))[1],
  gene_id = "id",
  add_markdown_panel = FALSE
)
```

Arguments

`sce`	SingleCellExperiment object
`features`	A character vector or a data.frame containing a list of genes. If `features` is a data.frame, the column containing the gene names must be named "id"
`reddim_type`	A string vector containing the dimensionality reduction type
`clusters`	A character string containing the name of the clusters/cell-type/state...(as listed in the colData of the sce)
`groups`	A character string of the groups/conditions...(as it appears in the colData of the sce)
`gene_id`	A character string containing the name of the column name containing gene names/ids, when 'features' is a data.frame
`add_markdown_panel`	A logical indicating whether or not to include the MarkdownBoard panel in the initial configuration

Value

A list of "Panel" objects specifying the initial state of iSEE instance

Examples

```
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
sce <- scater::runTSNE(sce)
gene_list <- c("ENSMUSG00000026581",
               "ENSMUSG00000005087",
               "ENSMUSG00000015437")
cluster <- "stimulus"
group <- "single cell quality"
initial <- iSEEinit(sce = sce, features = gene_list, clusters = cluster, groups = group)
```

`iSEEmarker` *iSEEmarker*

Description

`iSEEmarker()` creates an initial state of an iSEE instance for interactive exploration of marker genes through the `DynamicMarkerTable` panel, synchronizing selections with a `ReducedDimensionPlot` and a `FeatureAssayPlot` to visualize the expression of selected marker genes

Usage

```
iSEEmarker(
  sce,
  reddim_type = "TSNE",
  clusters = colnames(colData(sce))[1],
  groups = colnames(colData(sce))[1],
  selection_plot_format = c("ColumnDataPlot", "ReducedDimensionPlot")
)
```

Arguments

`sce`	SingleCellExperiment object
`reddim_type`	A string vector containing the dimensionality reduction
`clusters`	A character string containing the name of the clusters/cell-type/state...(as listed in the colData of the sce)
`groups`	A character string of the groups/conditions...(as it appears in the colData of the sce)
`selection_plot_format`	A string character containing the class of the panel. It can be either `ColumnDataPlot` or `ReducedDimensionPlot`

Value

A list of "Panel" objects specifying the initial state of iSEE instance

Examples

```
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
sce <- scater::runTSNE(sce)
cluster <- "stimulus"
group <- "single cell quality"
initial <- iSEEmarker(sce = sce, clusters = cluster, groups = group,
selection_plot_format = "ColumnDataPlot")
```

`iSEEnrich` *iSEEnrich*

Description

`iSEEnrich()` creates an initial state of an iSEE instance for interactive exploration of feature sets extracted from GO and KEGG database, displaying all associated genes in a `RowDataTable` panel.

Usage

```
iSEEnrich(
  sce,
  collection = c("GO", "KEGG"),
  organism = "org.Hs.eg.db",
  gene_identifier = "ENTREZID",
  clusters = colnames(colData(sce))[1],
  groups = colnames(colData(sce))[1],
  reddim_type = "PCA"
)
```

Arguments

`sce`	SingleCellExperiment object
`collection`	A character vector specifying the gene set collections of interest (GO,KEGG)
`organism`	A character string of the org.*.eg.db package to use to extract mappings of gene sets to gene IDs.
`gene_identifier`	A character string specifying the identifier to use to extract gene IDs for the organism package
`clusters`	A character string containing the name of the clusters/cell-type/state...(as listed in the colData of the sce)
`groups`	A character string of the groups/conditions...(as it appears in the colData of the sce)
`reddim_type`	A string vector containing the dimensionality reduction type

Value

A list of "Panel" objects specifying the initial state of iSEE instance

Examples

```
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
GO_collection <- "GO"
Mm_organism <- "org.Mm.eg.db"
gene_id <- "SYMBOL"
clusters <- "stimulus"
groups <- "single cell quality"
reddim_type <- "PCA"
results <- iSEEnrich(sce = sce,
                     collection = GO_collection,
                     organism = Mm_organism,
                     gene_identifier = gene_id,
                     clusters = clusters,
                     groups = groups,
                     reddim_type = reddim_type)
```

---

view_initial_network *View an initial object as a network*

Description

Translates the layout of the `initial` configuration object as a networks, representing panels as nodes and links between them as edges.

Usage

```
view_initial_network(initial, plot_format = c("igraph", "visNetwork", "none"))
```

Arguments

`initial`	An `initial` list object, in the format that is required to be passed as a parameter in the call to `iSEE::iSEE()`.
`plot_format`	Character string, one of `igraph`, `visNetwork`, or `none`. Defaults to `igraph`. Determines the format of the visual representation generated as a side effect of this function - it can be the output of the `plot()` function for `igraph` objects, or an interactive widget created via `visNetwork::visNetwork()`.

Details

Panels are the nodes, with color and names to identify them easily. The connections among panels are represented through directed edges. This can be a compact visualization to obtain an overview for the configuration, without the need of fully launching the app and loading the content of all panels

This function is particularly useful with mid-to-large `initial` objects, as they can be quickly generated in a programmatic manner via the `iSEEinit()` provided in this package.

Value

An `igraph` object, underlying the visual representation provided.

See Also

`view_initial_tiles()`

Examples

```
## Load a dataset and preprocess this quickly
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
sce <- scater::runTSNE(sce)
## Select some features and aspects to focus on
gene_list <- c("ENSMUSG00000026581", "ENSMUSG00000005087", "ENSMUSG00000015437")
cluster <- "stimulus"
group <- "single cell quality"
initial <- iSEEinit(sce = sce,
                    features = gene_list,
                    clusters = cluster,
                    groups = group)

g_init <- view_initial_network(initial)
g_init

view_initial_network(initial, plot_format = "visNetwork")

## Continue your exploration directly within iSEE!
if (interactive())
  iSEE(sce, initial = initial)
```

`view_initial_tiles` *View an initial object as a set of tiles*

Description

Previews the layout of the `initial` configuration object in a graphical form.

Usage

```
view_initial_tiles(initial)
```

Arguments

initial	An initial list object, in the format that is required to be passed as a parameter in the call to iSEE::iSEE().

Details

Tiles are used to represent the panel types, and reflect the values of their width. This can be a compact visualization to obtain an overview for the configuration, without the need of fully launching the app and loading the content of all panels

This function is particularly useful with mid-to-large initial objects, as they can be quickly generated in a programmatic manner via the iSEEinit() provided in this package.

Value

A ggplot object, representing a schematic view for the initial object.

See Also

view_initial_network()

Examples

```
## Load a dataset and preprocess this quickly
sce <- scRNAseq::RichardTCellData()
sce <- scuttle::logNormCounts(sce)
sce <- scater::runPCA(sce)
sce <- scater::runTSNE(sce)
## Select some features and aspects to focus on
gene_list <- c("ENSMUSG00000026581",
               "ENSMUSG00000005087",
               "ENSMUSG00000015437")
cluster <- "stimulus"
group <- "single cell quality"
initial <- iSEEinit(sce = sce,
                    features = gene_list,
                    clusters = cluster,
                    groups = group)

view_initial_tiles (initial)

## Continue your exploration directly within iSEE!
if (interactive())
  iSEE(sce, initial = initial)
```

Index

∗ **datasets**
constants-iSEEfier, 2
∗ **internal**
iSEEfier-pkg, 4

constants-iSEEfier, 2

glue_initials, 3

iSEE::iSEE(), *3*, *8*, *10*
iSEE_panel_colors *(*constants-iSEEfier*)*, 2
iSEEfier *(*iSEEfier-pkg*)*, 4
iSEEfier-package *(*iSEEfier-pkg*)*, 4
iSEEfier-pkg, 4
iSEEinit, 5
iSEEmarker, 6
iSEEnrich, 7

view_initial_network, 8
view_initial_network(), *10*
view_initial_tiles, 9
view_initial_tiles(), *9*