

Package ‘Dune’

March 30, 2021

Title Improving replicability in single-cell RNA-Seq cell type discovery

Version 1.2.0

Description Given a set of clustering labels, Dune merges pairs of clusters to increase mean ARI between labels, improving replicability.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

Imports BiocParallel, SummarizedExperiment, mclust, utils, ggplot2, dplyr, tidyr, RColorBrewer, magrittr, ganimate, purrr

Suggests knitr, rmarkdown, testthat (>= 2.1.0)

VignetteBuilder knitr

Depends R (>= 4.0)

biocViews Clustering, GeneExpression, RNASeq, Software, SingleCell, Transcriptomics, Visualization

git_url <https://git.bioconductor.org/packages/Dune>

git_branch RELEASE_3_12

git_last_commit a60b7f3

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

Author Hector Roux de Bezieux [aut, cre]
(<<https://orcid.org/0000-0002-1489-8339>>),
Kelly Street [aut]

Maintainer Hector Roux de Bezieux <hector.rouxdebezieux@berkeley.edu>

R topics documented:

.adjustedRandIndex	2
ARIEvolution	2
ARIImp	3
ARIs	4

ARItrend	5
clusMat	5
clusterConversion	6
ConfusionEvolution	6
ConfusionPlot	7
Dune	8
functionTracking	9
intermediateMat	10
nuclei	11
plotARIs	11
plotPrePost	12
whenToStop	12

Index	14
--------------	-----------

<code>.adjustedRandIndex</code>	<i>adjustedRandIndex</i>
---------------------------------	--------------------------

Description

`adjustedRandIndex`

Usage

`.adjustedRandIndex(tab)`

Arguments

`tab` The confusion matrix

Details

Second part of the [adjustedRandIndex](#) function from `mcLust`.

Value

The ARI

ARIEvolution	<i>Plot the evolution of the pairwise ARIs as merging happens</i>
--------------	---

Description

Animated version of [plotARIs](#)

Usage

```

ARIEvolution(
  merger,
  unclustered = NULL,
  values = TRUE,
  numericalLabels = FALSE,
  state_length = 1
)

```

Arguments

merger the result from having run [Dune](#) on the dataset

unclustered The value assigned to unclustered cells. Default to NULL

values Whether to also display the ARI values. Default to TRUE.

numericalLabels Whether labels are numerical values. Default to FALSE.

state_length Time between steps. Default to 1. See [transition_states](#) for details.

Details

See [plotARIs](#) and [animate](#).

Value

a gganim object

Examples

```

## Not run:
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
ARIEvolution(merger)
## End(Not run)

```

ARIImp

ARI improvement

Description

Compute the ARI improvement over the ARI merging procedure

Usage

```
ARIImp(merger, unclustered = NULL)
```

Arguments

merger the result from having run [Dune](#) on the dataset

unclustered The value assigned to unclustered cells. Default to NULL

Value

a vector with the mean ARI between methods at each merge

See Also

ARItrend

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
plot(0:nrow(merger$merges), ARIImp(merger))
```

ARIs

ARI Matrix

Description

ARI Matrix

Usage

```
ARIs(clusMat, unclustered = NULL)
```

Arguments

clusMat	The clustering matrix with a row per cell and a column per clustering label type
unclustered	The value assigned to unclustered cells. Default to NULL

Details

In the ARI matrix where each cell i,j is the adjusted Rand Index between columns i and j of the original clusMat. If unclustered is not NULL, the cells which have been assigned to the unclustered cluster will not be counted towards computing the ARI.

Value

The ARI matrix

Examples

```
data("clusMat", package = "Dune")
ARIs(clusMat)
```

ARItrend	<i>ARI improvement plot</i>
----------	-----------------------------

Description

A plot to see how ARI improves over merging

Usage

```
ARItrend(merger, unclustered = NULL)
```

Arguments

merger	the result from having run Dune on the dataset
unclustered	The value assigned to unclustered cells. Default to NULL

Value

a [ggplot](#) object

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
ARItrend(merger)
```

clusMat	<i>A clustering matrix used to demonstrate the ari-merging process.</i>
---------	---

Description

A clustering matrix used to demonstrate the ari-merging process.

Usage

```
clusMat
```

Format

An object of class `matrix` with 100 rows and 5 columns.

Details

This matrix has 100 samples with 5 cluster labels. Cluster labels 2 through 5 are modified versions of cluster label 1, where some clusters from label 1 were broken down into smaller clusters. It is just a toy dataset that can be re-generated with the code in https://github.com/HectorRDB/Pipeline_Brain/blob/master/Sandbox.

clusterConversion	<i>clusterConversion</i>
-------------------	--------------------------

Description

Find the conversion between the old cluster and the final clusters

Usage

```
clusterConversion(merger, p = 1, n_steps = NULL)
```

Arguments

merger	the result from having run Dune on the dataset
p	A value between 0 and 1. We stop when the mean ARI has improved by p of the final total improvement. Default to 1 (i.e running the full merging).
n_steps	Alternatively, you can specify the number of merging steps to do before stopping.

Value

A list containing a matrix per clustering method, with a column for the old labels and a column for the new labels.

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
clusterConversion(merger)[[2]]
```

ConfusionEvolution	<i>Plot the evolution of the ConfusionPlot as merging happens</i>
--------------------	---

Description

Animated version of [ConfusionPlot](#)

Usage

```
ConfusionEvolution(merger, unclustered = NULL, x, y, state_length = 1)
```

Arguments

merger	the result from having run Dune on the dataset
unclustered	The value assigned to unclustered cells. Default to NULL
x	The name of the first cluster label to plot
y	The name of the second cluster label to plot
state_length	Time between steps. Default to 1. See transition_states for details.

Details

See [ConfusionPlot](#) and [animate](#).

Value

a `gganim` object

Examples

```
## Not run:  
data("clusMat", package = "Dune")  
merger <- Dune(clusMat = clusMat)  
ConfusionEvolution(merger, x = "A", y = "B")  
## End(Not run)
```

ConfusionPlot	<i>Plot confusion matrix</i>
---------------	------------------------------

Description

A plot to visualize how alike two clustering labels are

Usage

```
ConfusionPlot(x, y = NULL)
```

Arguments

`x` A vector of clustering labels or a matrix of clustering labels. See details.
`y` Optional. Another vector of clustering labels

Value

a `ggplot` object

Examples

```
data("nuclei", package = "Dune")  
ConfusionPlot(nuclei[, c("SC3", "Monocle")])
```

Dune

Dune

Description

Compute the ARI between every pair of clustering labels after merging every possible pair of clusters. Find the one that improves the ARI merging the most, merge the pair. Repeat until there is no improvement.

Usage

```
Dune(clusMat, ...)

## S4 method for signature 'matrix'
Dune(
  clusMat,
  unclustered = NULL,
  verbose = FALSE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'data.frame'
Dune(
  clusMat,
  unclustered = NULL,
  verbose = FALSE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'SummarizedExperiment'
Dune(
  clusMat,
  cluster_columns,
  unclustered = NULL,
  verbose = FALSE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

<code>clusMat</code>	the matrix of samples by clustering labels.
<code>...</code>	parameters including:
<code>unclustered</code>	The value assigned to unclustered cells. Default to NULL
<code>verbose</code>	Whether or not the print cluster merging as it happens.
<code>parallel</code>	Logical, defaults to FALSE. Set to TRUE if you want to parallelize the fitting.

BPPARAM object of class `bpparamClass` that specifies the back-end to be used for computations. See `bpparam` in `BiocParallel` package for details. Won't be used if `parallel` is `FALSE`.

cluster_columns if `clusMat` is a [SummarizedExperiment](#), then this defines the columns of `colData` that are outputs from a clustering algorithm.

Details

The Dune algorithm merges pairs of clusters in order to improve the mean adjusted Rand Index with other clustering labels. It returns a list with four components.: #'

- `initialMat`: The initial matrix of cluster labels
- `currentMat`: The final matrix of cluster labels
- `merges`: The step-by-step detail of the merges, recapitulating which clusters where merged in which cluster label
- `impARI`: How much each merge improved the mean ARI between the cluster label that has been merged and the other cluster labels.

Value

A list with four components: the initial matrix of clustering labels, the final matrix of clustering labels, the merge info matrix and the ARI improvement vector.

See Also

`clusterConversion` `ARIImp`

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
# clusters 11 to 14 from cluster label 5 and 3 are subset of cluster 2 from
# other cluster labels. Designing cluster 2 as unclustered therefore means we
# do fewer merges.
merger2 <- Dune(clusMat = clusMat, unclustered = 2)
merger$merges
merger2$merges
```

`functionTracking` *Track the evolution of a function along merging*

Description

For a given ARI merging, compute the evolution on the function `f`

Usage

```
functionTracking(merger, f, p = 1, n_steps = NULL, ...)
```

Arguments

merger	the result from having run Dune on the dataset
f	the function used. It must takes as input a clustering matrix and return a value
p	A value between 0 and 1. We stop when the mean ARI has improved by p of the final total improvement. Default to 1 (i.e running the full merging).
n_steps	Alternatively, you can specify the number of merging steps to do before stopping.
...	additional arguments passed to f

Value

a vector of length the number of merges

Examples

```
# Return the number of clusters for the fourth cluster label
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
f <- function(clusMat, i) dplyr::n_distinct(clusMat[, i])
functionTracking(merger, f, i = 4)
```

intermediateMat	<i>Find the clustering matrix that we would get if we stopped the ARI merging early</i>
-----------------	---

Description

Find the clustering matrix that we would get if we stopped the ARI merging early

Usage

```
intermediateMat(merger, p = 1, n_steps = NULL)
```

Arguments

merger	the result from having run Dune on the dataset
p	A value between 0 and 1. We stop when the mean ARI has improved by p of the final total improvement. Default to 1 (i.e running the full merging).
n_steps	Alternatively, you can specify the number of merging steps to do before stopping.

Value

A matrix with the same dimensions as the currentmMat of the merger argument

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
head(intermediateMat(merger, n_steps = 1))
```

nuclei	<i>Cluster labels for a subset of the allen Smart-Seq nuclei dataset</i>
--------	--

Description

Cluster labels for a subset of the allen Smart-Seq nuclei dataset

Usage

```
nuclei
```

Format

An object of class `data.frame` with 1744 rows and 7 columns.

Details

This matrix of clusters was obtained by running 3 clustering algorithms on a brain snRNA-Seq dataset from Tasic et al (<https://doi.org/10.1038/s41586-018-0654-5>). This dataset was then sub-setted to the GABAergic neurons. Code to reproduce all this can be found in the github repository from the Dune paper (https://github.com/HectorRDB/Dune_Paper).

plotARIs	<i>Plot an heatmap of the ARI matrix</i>
----------	--

Description

We can compute the ARI between pairs of cluster labels. This function plots a matrix where a cell is the adjusted Rand Index between cluster label of row *i* and cluster label of column *j*.

Usage

```
plotARIs(clusMat, unclustered = NULL, values = TRUE, numericalLabels = FALSE)
```

Arguments

<code>clusMat</code>	The clustering matrix with a row per cell and a column per clustering label type
<code>unclustered</code>	The value assigned to unclustered cells. Default to <code>NULL</code>
<code>values</code>	Whether to also display the ARI values. Default to <code>TRUE</code> .
<code>numericalLabels</code>	Whether labels are numerical values. Default to <code>FALSE</code> .

Value

a `ggplot` object

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
plotARIs(merger$initialMat)
plotARIs(merger$currentMat)
```

 plotPrePost

Plot the reduction in cluster size for an ARI merging with Dune

Description

Plot the reduction in cluster size for an ARI merging with Dune

Usage

```
plotPrePost(merger)
```

Arguments

merger The output from an ARI merging, by calling [Dune](#)

Value

a [ggplot](#) object #' @importFrom dplyr mutate

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
plotPrePost(merger)
```

whenToStop

When to Stop

Description

When to Stop

Usage

```
whenToStop(merger, p)
```

Arguments

merger the result from having run [Dune](#) on the dataset

p A value between 0 and 1. We stop when the mean ARI has improved by p of the final total improvement. Default to 1 (i.e running the full merging).

Details

The [Dune](#) process improves the mean ARI. This return the first merging step after which the mean ARI has been improved by p of the total. Setting p = 1 just return the number of merges.

Value

An integer giving the step where to stop.

Examples

```
data("clusMat", package = "Dune")
merger <- Dune(clusMat = clusMat)
whenToStop(merger, p = .5)
```

Index

- * **datasets**
 - clusMat, 5
 - nuclei, 11
- .adjustedRandIndex, 2
- adjustedRandIndex, 2
- animate, 3, 7
- ARIEvolution, 2
- ARIImp, 3
- ARIs, 4
- ARItrend, 5

- clusMat, 5
- clusterConversion, 6
- ConfusionEvolution, 6
- ConfusionPlot, 6, 7, 7

- Dune, 3, 5, 6, 8, 10, 12
- Dune, data.frame-method (Dune), 8
- Dune, matrix-method (Dune), 8
- Dune, SummarizedExperiment-method (Dune), 8

- functionTracking, 9

- ggplot, 5, 7, 11, 12

- intermediateMat, 10

- mclust, 2

- nuclei, 11

- plotARIs, 2, 3, 11
- plotPrePost, 12

- SummarizedExperiment, 9

- transition_states, 3, 6

- whenToStop, 12