

Package ‘scimo’

July 23, 2025

Title Extra Recipes Steps for Dealing with Omics Data

Version 0.0.2

Description Omics data (e.g. transcriptomics, proteomics, metagenomics...) offer a detailed and multi-dimensional perspective on the molecular components and interactions within complex biological (eco)systems. Analyzing these data requires adapted procedures, which are implemented as steps according to the 'recipes' package.

License GPL (>= 3)

URL <https://github.com/abichat/scimo>

BugReports <https://github.com/abichat/scimo/issues>

Depends R (>= 2.10), recipes

Imports dplyr, generics, magrittr, rlang, stats, tibble, tidyr

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData false

RoxygenNote 7.3.1

NeedsCompilation no

Author Antoine BICHAT [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6599-7081>>),
Julie AUBERT [ctb] (ORCID: <<https://orcid.org/0000-0001-5203-5748>>)

Maintainer Antoine BICHAT <antoine.bichat@proton.me>

Repository CRAN

Date/Publication 2024-06-07 17:40:02 UTC

Contents

cheese_abundance	2
pedcan_expression	3
step_aggregate_hclust	4
step_aggregate_list	5
step_rownormalize_tss	7
step_select_background	8
step_select_cv	10
step_select_kruskal	11
step_select_wilcoxon	13
step_taxonomy	14
Index	17

cheese_abundance	<i>Abundance of Fungal Communities in Cheese</i>
------------------	--

Description

Fungal community abundance of 74 ASVs sampled from the surface of three different French cheeses.

Usage

```
data("cheese_abundance", package = "scimo")

data("cheese_taxonomy", package = "scimo")
```

Format

For cheese_abundance, a [tibble](#) with columns:

sample Sample ID.

cheese Appellation of the cheese. One of Saint-Nectaire, Livarot or Epoisses.

rind_type One of Natural or Washed.

other columns Count of the ASV.

For cheese_taxonomy, a [tibble](#) with columns:

asv Amplicon Sequence Variant (ASV) ID.

lineage Character corresponding to a standard concatenation of taxonomic clades.

other columns Clade to which the ASV belongs.

Source

This dataset came from [doi:10.24072/pcjournal.321](https://doi.org/10.24072/pcjournal.321).

Examples

```
data("cheese_abundance", package = "scimo")
cheese_abundance
data("cheese_taxonomy", package = "scimo")
cheese_taxonomy
```

pedcan_expression	<i>Gene Expression of Pediatric Cancer</i>
-------------------	--

Description

Gene expression of 108 CCLE cell lines from 5 different pediatric cancers.

Usage

```
data("pedcan_expression", package = "scimo")
```

Format

A [tibble](#) with columns:

cell_line Cell line name.

sex One of Male, Female or Unknown.

event One of Primary, Metastasis or Unknown.

disease One of Neuroblastoma, Ewing Sarcoma, Rhabdomyosarcoma, Embryonal Tumor or Osteosarcoma.

other columns Expression of the gene, given in $\log_2(\text{TPM} + 1)$.

Source

This dataset is generated from DepMap Public 23Q4 primary files. <https://depmap.org/portal/download/all/>.

Examples

```
data("pedcan_expression", package = "scimo")
pedcan_expression
```

step_aggregate_hclust *Feature aggregation step based on a hierarchical clustering*

Description

Aggregate variables according to hierarchical clustering.

Usage

```
step_aggregate_hclust(
  recipe,
  ...,
  role = "predictor",
  trained = FALSE,
  n_clusters,
  fun_agg,
  dist_metric = "euclidean",
  linkage_method = "complete",
  res = NULL,
  prefix = "cl_",
  keep_original_cols = FALSE,
  skip = FALSE,
  id = rand_id("aggregate_hclust")
)

## S3 method for class 'step_aggregate_hclust'
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	For model terms created by this step, what analysis role should they be assigned? By default, the new columns created by this step from the original variables will be used as predictors in a model.
trained	A logical to indicate if the quantities for preprocessing have been estimated.
n_clusters	Number of cluster to create.
fun_agg	Aggregation function like sum or mean.
dist_metric	Default to euclidean. See stats::dist() for more details.
linkage_method	Default to complete. See stats::hclust() for more details.
res	This parameter is only produced after the recipe has been trained.
prefix	A character string for the prefix of the resulting new variables.

keep_original_cols	A logical to keep the original variables in the output. Defaults to FALSE.
skip	A logical. Should the step be skipped when the recipe is baked by <code>bake()</code> ? While all operations are baked when <code>prep()</code> is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using <code>skip = TRUE</code> as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A <code>step_aggregate_hclust</code> object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
rec <-
  iris %>%
  recipe(formula = Species ~ .) %>%
  step_aggregate_hclust(all_numeric_predictors(),
                        n_clusters = 2, fun_agg = sum) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)
```

step_aggregate_list	<i>Feature aggregation step based on a defined list</i>
---------------------	---

Description

Aggregate variables according to prior knowledge.

Usage

```
step_aggregate_list(
  recipe,
  ...,
  role = "predictor",
  trained = FALSE,
  list_agg = NULL,
  fun_agg = NULL,
  others = "discard",
  name_others = "others",
```

```

    res = NULL,
    prefix = "agg_",
    keep_original_cols = FALSE,
    skip = FALSE,
    id = rand_id("aggregate_list")
  )

## S3 method for class 'step_aggregate_list'
tidy(x, ...)

```

Arguments

<code>recipe</code>	A recipe object. The step will be added to the sequence of operations for this recipe.
<code>...</code>	One or more selector functions to choose variables for this step. See selections() for more details.
<code>role</code>	For model terms created by this step, what analysis role should they be assigned? By default, the new columns created by this step from the original variables will be used as predictors in a model.
<code>trained</code>	A logical to indicate if the quantities for preprocessing have been estimated.
<code>list_agg</code>	Named list of aggregated variables.
<code>fun_agg</code>	Aggregation function like sum or mean.
<code>others</code>	Behavior for the selected variables in <code>...</code> that are not present in <code>list_agg</code> . If <code>discard</code> (the default), they are not kept. If <code>asis</code> , they are kept without modification. If <code>aggregate</code> , they are aggregated in a new variable.
<code>name_others</code>	If <code>others</code> is set to <code>aggregate</code> , name of the aggregated variable. Not used otherwise.
<code>res</code>	This parameter is only produced after the recipe has been trained.
<code>prefix</code>	A character string for the prefix of the resulting new variables that are not named in <code>list_agg</code> .
<code>keep_original_cols</code>	A logical to keep the original variables in the output. Defaults to <code>FALSE</code> .
<code>skip</code>	A logical. Should the step be skipped when the recipe is baked by bake() ? While all operations are baked when prep() is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using <code>skip = TRUE</code> as it may affect the computations for subsequent operations.
<code>id</code>	A character string that is unique to this step to identify it.
<code>x</code>	A <code>step_aggregate_list</code> object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
list_iris <- list(sepal.size = c("Sepal.Length", "Sepal.Width"),
                 petal.size = c("Petal.Length", "Petal.Width"))

rec <-
  iris %>%
  recipe(formula = Species ~ .) %>%
  step_aggregate_list(all_numeric_predictors(),
                     list_agg = list_iris, fun_agg = prod) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)
```

step_rownormalize_tss *Feature normalization step using total sum scaling*

Description

Normalize a set of variables by converting them to proportion, making them sum to 1. Also known as simplex projection.

Usage

```
step_rownormalize_tss(
  recipe,
  ...,
  role = NA,
  trained = FALSE,
  res = NULL,
  skip = FALSE,
  id = rand_id("rownormalize_tss")
)

## S3 method for class 'step_rownormalize_tss'
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	Not used by this step since no new variables are created.

trained	A logical to indicate if the quantities for preprocessing have been estimated.
res	This parameter is only produced after the recipe has been trained.
skip	A logical. Should the step be skipped when the recipe is baked by <code>bake()</code> ? While all operations are baked when <code>prep()</code> is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using <code>skip = TRUE</code> as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A <code>step_rownormalize_tss</code> object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
rec <-
  recipe(Species ~ ., data = iris) %>%
  step_rownormalize_tss(all_numeric_predictors()) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)
```

step_select_background

Feature selection step using background level

Description

Select features that exceed a background level in at least a defined number of samples.

Usage

```
step_select_background(
  recipe,
  ...,
  role = NA,
  trained = FALSE,
  background_level = NULL,
  n_samples = NULL,
  prop_samples = NULL,
  res = NULL,
```

```

    skip = FALSE,
    id = rand_id("select_background")
  )

## S3 method for class 'step_select_background'
tidy(x, ...)

```

Arguments

<code>recipe</code>	A recipe object. The step will be added to the sequence of operations for this recipe.
<code>...</code>	One or more selector functions to choose variables for this step. See selections() for more details.
<code>role</code>	Not used by this step since no new variables are created.
<code>trained</code>	A logical to indicate if the quantities for preprocessing have been estimated.
<code>background_level</code>	Background level to exceed.
<code>n_samples, prop_samples</code>	Count or proportion of samples in which a feature exceeds <code>background_level</code> to be retained.
<code>res</code>	This parameter is only produced after the recipe has been trained.
<code>skip</code>	A logical. Should the step be skipped when the recipe is baked by bake() ? While all operations are baked when prep() is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using <code>skip = TRUE</code> as it may affect the computations for subsequent operations.
<code>id</code>	A character string that is unique to this step to identify it.
<code>x</code>	A <code>step_select_background</code> object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```

rec <-
  iris %>%
  recipe(formula = Species ~ .) %>%
  step_select_background(all_numeric_predictors(),
                        background_level = 4, prop_samples = 0.5) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)

```

step_select_cv	<i>Feature selection step using the coefficient of variation</i>
----------------	--

Description

Select variables with highest coefficient of variation.

Usage

```
step_select_cv(
  recipe,
  ...,
  role = NA,
  trained = FALSE,
  n_kept = NULL,
  prop_kept = NULL,
  cutoff = NULL,
  res = NULL,
  skip = FALSE,
  id = rand_id("select_cv")
)

## S3 method for class 'step_select_cv'
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	Not used by this step since no new variables are created.
trained	A logical to indicate if the quantities for preprocessing have been estimated.
n_kept	Number of variables to keep.
prop_kept	A numeric value between 0 and 1 representing the proportion of variables to keep. n_kept and prop_kept are mutually exclusive.
cutoff	Threshold beyond which (below or above) the variables are discarded.
res	This parameter is only produced after the recipe has been trained.
skip	A logical. Should the step be skipped when the recipe is baked by bake() ? While all operations are baked when prep() is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using skip = TRUE as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A step_select_cv object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
rec <-  
  recipe(Species ~ ., data = iris) %>%  
  step_select_cv(all_numeric_predictors(), n_kept = 2) %>%  
  prep()  
rec  
tidy(rec, 1)  
bake(rec, new_data = NULL)
```

step_select_kruskal	<i>Feature selection step using Kruskal test</i>
---------------------	--

Description

Select variables with the lowest (adjusted) p-value of a Kruskal-Wallis test against an outcome.

Usage

```
step_select_kruskal(  
  recipe,  
  ...,  
  role = NA,  
  trained = FALSE,  
  outcome = NULL,  
  n_kept = NULL,  
  prop_kept = NULL,  
  cutoff = NULL,  
  correction = "none",  
  res = NULL,  
  skip = FALSE,  
  id = rand_id("select_kruskal")  
)  
  
## S3 method for class 'step_select_kruskal'  
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	Not used by this step since no new variables are created.
trained	A logical to indicate if the quantities for preprocessing have been estimated.
outcome	Name of the variable to perform the test against.
n_kept	Number of variables to keep.
prop_kept	A numeric value between 0 and 1 representing the proportion of variables to keep. n_kept and prop_kept are mutually exclusive.
cutoff	Threshold beyond which (below or above) the variables are discarded.
correction	Multiple testing correction method. One of p.adjust.methods. Default to "none".
res	This parameter is only produced after the recipe has been trained.
skip	A logical. Should the step be skipped when the recipe is baked by bake() ? While all operations are baked when prep() is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using skip = TRUE as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A step_select_kruskal object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
rec <-
  iris %>%
  recipe(formula = Species ~ .) %>%
  step_select_kruskal(all_numeric_predictors(), outcome = "Species",
                      correction = "fdr", prop_kept = 0.5) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)
```

step_select_wilcoxon *Feature selection step using Wilcoxon test*

Description

Select variables with the lowest (adjusted) p-value of a Wilcoxon-Mann-Whitney test against an outcome.

Usage

```
step_select_wilcoxon(
  recipe,
  ...,
  role = NA,
  trained = FALSE,
  outcome = NULL,
  n_kept = NULL,
  prop_kept = NULL,
  cutoff = NULL,
  correction = "none",
  res = NULL,
  skip = FALSE,
  id = rand_id("select_wilcoxon")
)

## S3 method for class 'step_select_wilcoxon'
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	Not used by this step since no new variables are created.
trained	A logical to indicate if the quantities for preprocessing have been estimated.
outcome	Name of the variable to perform the test against.
n_kept	Number of variables to keep.
prop_kept	A numeric value between 0 and 1 representing the proportion of variables to keep. n_kept and prop_kept are mutually exclusive.
cutoff	Threshold beyond which (below or above) the variables are discarded.
correction	Multiple testing correction method. One of p.adjust.methods. Default to "none".
res	This parameter is only produced after the recipe has been trained.

skip	A logical. Should the step be skipped when the recipe is baked by <code>bake()</code> ? While all operations are baked when <code>prep()</code> is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using <code>skip = TRUE</code> as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A <code>step_select_wilcoxon</code> object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
rec <-
  iris %>%
  dplyr::filter(Species != "virginica") %>%
  recipe(formula = Species ~ .) %>%
  step_select_wilcoxon(all_numeric_predictors(), outcome = "Species",
                      correction = "fdr", prop_kept = 0.5) %>%
  prep()
rec
tidy(rec, 1)
bake(rec, new_data = NULL)
```

step_taxonomy

Taxonomic clades feature generator

Description

Extract clades from a lineage, as defined in the `{yatah}` package.

Usage

```
step_taxonomy(
  recipe,
  ...,
  role = "predictor",
  trained = FALSE,
  rank = NULL,
  res = NULL,
  keep_original_cols = FALSE,
  skip = FALSE,
  id = rand_id("taxonomy")
```

```
)

## S3 method for class 'step_taxonomy'
tidy(x, ...)
```

Arguments

recipe	A recipe object. The step will be added to the sequence of operations for this recipe.
...	One or more selector functions to choose variables for this step. See selections() for more details.
role	For model terms created by this step, what analysis role should they be assigned? By default, the new columns created by this step from the original variables will be used as predictors in a model.
trained	A logical to indicate if the quantities for preprocessing have been estimated.
rank	The desired ranks, a combinaison of "kingdom", "phylum", "class", "order", "family", "genus", "species", or "strain". See yatah::get_clade() for more details.
res	This parameter is only produced after the recipe has been trained.
keep_original_cols	A logical to keep the original variables in the output. Defaults to FALSE.
skip	A logical. Should the step be skipped when the recipe is baked by bake() ? While all operations are baked when prep() is run, some operations may not be able to be conducted on new data (e.g. processing the outcome variable(s)). Care should be taken when using skip = TRUE as it may affect the computations for subsequent operations.
id	A character string that is unique to this step to identify it.
x	A step_taxonomy object.

Value

An updated version of recipe with the new step added to the sequence of any existing operations.

Author(s)

Antoine Bichat

Examples

```
data("cheese_taxonomy")
rec <-
  cheese_taxonomy %>%
  select(asv, lineage) %>%
  recipe(~ .) %>%
  step_taxonomy(lineage, rank = c("order", "genus")) %>%
  prep()
rec
```

```
tidy(rec, 1)  
bake(rec, new_data = NULL)
```

Index

* datasets

- cheese_abundance, [2](#)
- pedcan_expression, [3](#)

bake(), [5](#), [6](#), [8–10](#), [12](#), [14](#), [15](#)

cheese_abundance, [2](#)
cheese_taxonomy (cheese_abundance), [2](#)

pedcan_expression, [3](#)
prep(), [5](#), [6](#), [8–10](#), [12](#), [14](#), [15](#)

selections(), [4](#), [6](#), [7](#), [9](#), [10](#), [12](#), [13](#), [15](#)
stats::dist(), [4](#)
stats::hclust(), [4](#)
step_aggregate_hclust, [4](#)
step_aggregate_list, [5](#)
step_rownormalize_tss, [7](#)
step_select_background, [8](#)
step_select_cv, [10](#)
step_select_kruskal, [11](#)
step_select_wilcoxon, [13](#)
step_taxonomy, [14](#)

tibble, [2](#), [3](#)
tidy.step_aggregate_hclust
 (step_aggregate_hclust), [4](#)
tidy.step_aggregate_list
 (step_aggregate_list), [5](#)
tidy.step_rownormalize_tss
 (step_rownormalize_tss), [7](#)
tidy.step_select_background
 (step_select_background), [8](#)
tidy.step_select_cv (step_select_cv), [10](#)
tidy.step_select_kruskal
 (step_select_kruskal), [11](#)
tidy.step_select_wilcoxon
 (step_select_wilcoxon), [13](#)
tidy.step_taxonomy (step_taxonomy), [14](#)

yatah::get_clade(), [15](#)