

Package: reclanc (via r-universe)

March 3, 2025

Type Package

Title A Revival of the ClaNC Algorithm

Version 0.0.0.9000

Description Classification of microarrays to nearest centroids (ClaNC) [doi:10.1093/bioinformatics/bti756](https://doi.org/10.1093/bioinformatics/bti756) selects optimal genes for centroids, similar to Prediction Analysis for Microarrays (PAM) but using fewer corrective factors, resulting in greater sensitivity and accuracy. Unfortunately, the original source of ClaNC can no longer be found. 'reclanc' reimplements this algorithm, with the the additional benefit of increased interoperability with standard data structures and modeling ecosystems.

License MIT + file LICENSE

URL <https://github.com/KaiAragaki/reclanc>,
<https://kaiaragaki.github.io/reclanc/>

BugReports <https://github.com/KaiAragaki/reclanc/issues>

Depends R (>= 4.1.0)

Encoding UTF-8

Language en

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests aws.s3, Biobase, celebrate, knitr, recipes, rmarkdown,
S4Vectors, SummarizedExperiment, testthat (>= 3.0.0),
tidymodels

Remotes McConkeyLab/celebrate

Config/testthat/edition 3

Imports cli, collapse, dplyr, hardhat, parsnip, rlang, stats, tibble

LazyData true

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev

Repository <https://kaiaragaki.r-universe.dev>

RemoteUrl <https://github.com/KaiAragaki/reclanc>

RemoteRef HEAD

RemoteSha 4285683e6cddfc50dd12927cd5ae564bd00ae9bb

Contents

clanc	2
predict.clanc	5
synthetic_expression	6
Index	7

clanc	<i>Calculate centroids from expression data with ClaNC</i>
-------	--

Description

Calculate centroids from expression data with ClaNC

Usage

```
clanc(x, ...)

## Default S3 method:
clanc(x, ...)

## S3 method for class 'data.frame'
clanc(x, classes, active, priors = "equal", ...)

## S3 method for class 'matrix'
clanc(x, classes, active, priors = "equal", ...)

## S3 method for class 'SummarizedExperiment'
clanc(x, classes, active, priors = "equal", assay = 1, ...)

## S3 method for class 'ExpressionSet'
clanc(x, classes, active, priors = "equal", ...)

## S3 method for class 'formula'
clanc(formula, data, active, priors = "equal", ...)

## S3 method for class 'recipe'
clanc(x, data, active, priors = "equal", ...)
```

Arguments

x	<p>Depending on the context:</p> <ul style="list-style-type: none"> • A data frame of expression. • A matrix of expression. • A recipe specifying a set of preprocessing steps created from <code>recipes::recipe()</code>. • An ExpressionSet. • A SummarizedExperiment with assay containing expression. <p>Expression should be library-size corrected, but not scaled.</p> <p>If supplying a data frame, matrix, ExpressionSet, SummarizedExperiment, the rows should represent genes, and the columns should represent samples (as is standard for expression data). The column names should be sample IDs, while the row names should be gene IDs.</p> <p>If a recipe is provided, the data should have genes as columns (to match the formula provided to the recipe.)</p>
...	Not currently used, but required for extensibility.
classes	<p>When x is a data frame or matrix, class contains class labels with the form of either:</p> <ul style="list-style-type: none"> • A data frame with 1 factor column • A factor vector. <p>When x is an ExpressionSet or SummarizedExperiment, class is the name of the column in <code>pData(x)</code> or <code>colData(x)</code> that contains classes as a factor.</p>
active	<p>Either a single number or a numeric vector equal to the length of the number of unique class labels. Represents the number class-specific genes that should be selected for a centroid. Note that different numbers of genes can be selected for each class. See details.</p> <p>When x is an ExpressionSet or SummarizedExperiment, active can additionally by the name of the column in <code>pData(x)</code> or <code>colData(x)</code> that contains the numeric vector</p>
priors	<p>Can take a variety of values:</p> <ul style="list-style-type: none"> • "equal" - each class has an equal prior • "class" - each class has a prior equal to its frequency in the training set • A numeric vector with length equal to number of classes <p>When x is an ExpressionSet or SummarizedExperiment, active can additionally by the name of the column in <code>pData(x)</code> or <code>colData(x)</code> that contains the numeric vector</p>
assay	When a SummarizedExperiment is used, the index or name of the assay
formula	A formula specifying the classes on the left-hand side, and the predictor terms on the right-hand side.
data	<p>When a recipe or formula is used, data is specified as:</p> <ul style="list-style-type: none"> • A data frame containing both expression and classes, where columns are the genes or class, and rows are the samples.

Details

The original description of ClaNC can be found [here](#)

While `active` sets the number of class-specific genes, each centroid will have more than that number of genes. To explain by way of example, if `active = 5` and there are 3 classes, each centroid will have 15 genes, with 5 of those genes being particular to a given class. If these genes are 'active' in that class, their values will be the mean of the class. If the genes are not active in that given class, their values will be the overall expression of the given gene across all classes.

Value

A `clanc` object.

Examples

```
expression_matrix <- synthetic_expression$expression
head(expression_matrix)
classes <- synthetic_expression$classes
classes

# data.frame/tibble/matrix interface:

clanc(expression_matrix, classes = classes, active = 5, priors = "equal")

# Formula interface:

# Data must have class included as a column
# Genes must be *columns* and samples must be *rows*
# Hence the data transposition.
for_formula <- data.frame(class = classes, t(expression_matrix))

clanc(class ~ ., for_formula, active = 5, priors = "equal")

# Recipes interface:

rec <- recipes::recipe(class ~ ., data = for_formula)

clanc(rec, for_formula, active = 5, priors = "equal")

# SummarizedExperiment interface:
se <- SummarizedExperiment::SummarizedExperiment(
  expression_matrix,
  colData = data.frame(
    class = classes,
    active = 5,
    prior = c(0.5, 0.5)
  )
)

clanc(se, classes = "class", active = "active", priors = "equal")
```

```

# ExpressionSet interface:
adf <- data.frame(
  row.names = colnames(expression_matrix),
  class = classes
) |>
  Biobase::AnnotatedDataFrame()

es <- Biobase::ExpressionSet(expression_matrix, adf)
clanc(es, classes = "class", active = 5, priors = 0.5)

```

predict.clanc	<i>Predict from a clanc</i>
---------------	-----------------------------

Description

Predict from a clanc

Usage

```

## S3 method for class 'clanc'
predict(object, new_data, type, assay = NULL, format = c("wide", "tall"), ...)

```

Arguments

object	A clanc object.
new_data	A data frame or matrix of new predictors.
type	A single character. The type of predictions to generate. Valid options are: <ul style="list-style-type: none"> "numeric" for numeric predictions.
assay	If object inherits SummarizedExperiment, the index of the assay.
format	Character. Are the data "wide" (default), with genes as columns, or "tall", with genes as rows?
...	Not used, but required for extensibility.
method	If type is numeric, the method of correlation

Value

A tibble of predictions. The number of rows in the tibble is guaranteed to be the same as the number of rows in new_data.

synthetic_expression *Synthetic Expression of Two Distinct Classes*

Description

Synthetic Expression of Two Distinct Classes

Usage

synthetic_expression

Format

synthetic_expression:

A list containing two items:

expression Normalized log expression of 12 samples across 100 genes

classes A factor vector of classes of each of the 12 samples

Index

* **datasets**

 synthetic_expression, 6

clanc, 2

predict.clanc, 5

recipes::recipe(), 3

synthetic_expression, 6