Introduction to the eCV package

Contents

Welcome!

Thanks for using package eCV! This vignette provides a high-level introduction to the functionality and usage of eCV and related functions.

Installation

To install the development version from GitHub, please do

```
# install.packages("remotes")
remotes::install_github("eclipsebio/eCV")
```

Load package eCV

Once installed, you can load the package eCV using `library`.

```
library("eCV")
Loading required package: idr
Loading required package: mvtnorm
Loading required package: future
Loading required package: future.apply
Loading required package: MatrixGenerics
Loading required package: matrixStats
```

```
Attaching package: 'MatrixGenerics'
The following objects are masked from 'package:matrixStats':

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars


 ____________________________________________________________________
|                          ______     __                             |
|                  ___    / ___\ \   / /                             |
|                 / _ \ | |     \ \ / /                              |
|                |  __/ | |___   \ V /                               |
|                 \___|  \____|   \_/                                |
|____________________________________________________________________|

        Enhanced Coefficient of Variation and IDR Extensions
                  for Reproducibility Assessment

This package provides extensions and alternative methods to  IDR  to
measure the reproducibility of omic data with an arbitrary number of
replicates. It introduces an enhanced Coefficient of Variation (eCV)
metric to assess the likelihood of omic features being reproducible.
```

Usage examples

The following examples assume you have installed and loaded package `tidyverse`.

```
library("tidyverse")
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.3     v purrr   1.0.2
v tibble  3.2.1     v dplyr   1.1.2
v tidyr   1.3.0     v stringr 1.5.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::count()  masks matrixStats::count()
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

Simulate data: function `simulate_data`

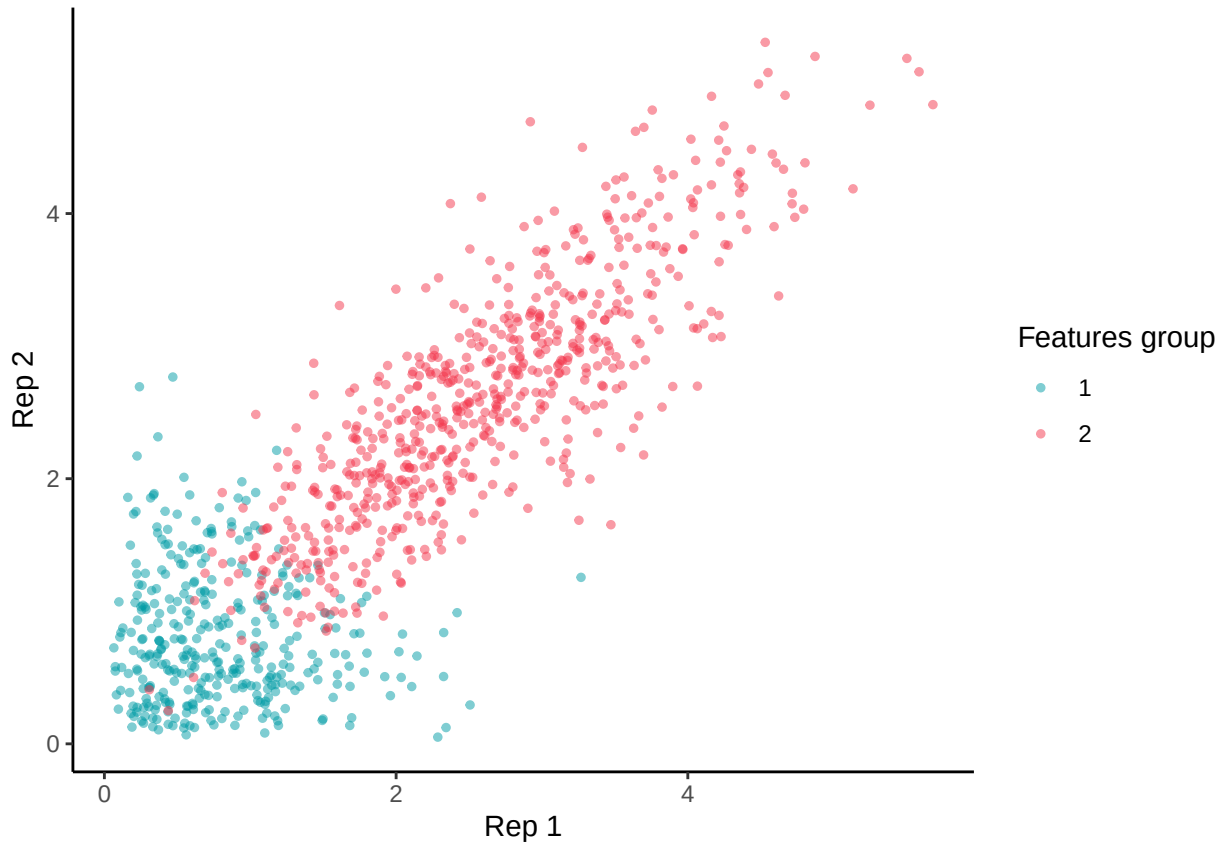
The `simulate_data` function simulates omic features, specifically designed to mimic reproducible and irreproducible groups. Building upon the copula mixture model simulations introduced in the original publication Li et al. (2011). This function produces samples of paired omic features for a specified number of replicates. The reproducibility state of each feature is determined by a binomial variable, where the mixing probabilities between two multivariate normal distributions are controlled by a vector of probabilities.

Users can tailor simulations using the scenario parameter, aligning with predefined scenarios from Li et al. (2011). These scenarios range from highly correlated and abundant reproducible features (Scenario 1) to situations with fewer, less correlated reproducible features (Scenarios 2 and 3), and even introduce a "reproducible noise" component (Scenario 4).

Users can customize the number of replicates (n_reps), the quantity of features (n_features), and the simulation scenario, while the function returns simulated data and associated parameter values in a convenient list format.

Here is an example that simulates 1,000 features across four replicates according to "Scenario 1".

```
set.seed(42)
out <- simulate_data(scenario = 1, n_reps = 4, n_features = 1000)
out$sim_data %>% as.data.frame() %>%
 mutate(`Features group` = as.character(out$sim_params$feature_group)) %>%
 ggplot(aes(x=`Rep 1`,y=`Rep 2`,color=`Features group`)) +
  geom_point(size=1, alpha=0.5) +
  scale_color_manual(values = c( "#009CA6" , "#F4364C")) +
  theme_classic()
```



Assessing feature reproducibility with different methods: function `mrep_assessment`

The `mrep_assessment` function is a comprehensive tool within the package eCV for evaluating the reproducibility of omic feature values across multiple sample replicates. This function acts as a wrapper for various methods, allowing users to choose between "IDR," "gIDR," "mIDR," and "eCV" based on their specific needs.

When employing the traditional "IDR" method, the function utilizes the `idr` package's implementation (`idr::est.IDR`), considering only the first two replicates regardless of the total available. For scenarios with more than two replicates, users can opt for "gIDR" and "mIDR," both of which reduce to the traditional IDR when there are only two replicates.

When calling the "eCV" method within `mrep_assessment`, an "enhanced" coefficient of variation (eCV) is calculated. This metric gauges the likelihood of an omic feature being reproducible by considering the variability among replicates and the mean intensity. The eCV is calculated as $\frac{|\sigma^2-\mu^2|}{\mu^2}$, offering insights into the noise (σ) and mean (μ) intensity characteristics of the features.

The "eCV" method further extends its analysis by making inferences based on the probabilities of eCV values originating from the group of reproducible features. It assumes a prior Normal distribution for reproducible features, sampling pseudo replicates using Probabilistic Bootstrap.

The user-friendly interface of mrep_assessment requires a numeric matrix `x` representing omic features across replicates, and allows customization through the `method` and `param` parameters.

The function returns a list with two elements: a numeric vector `rep_index` indicating reproducibility levels, and a string specifying the method used in this assessment. Additionally, users can leverage parallel computing via `future` and `future.apply` packages. This option is managed by adjusting the `n_threads` parameter for the "mIDR" and "eCV" methods.

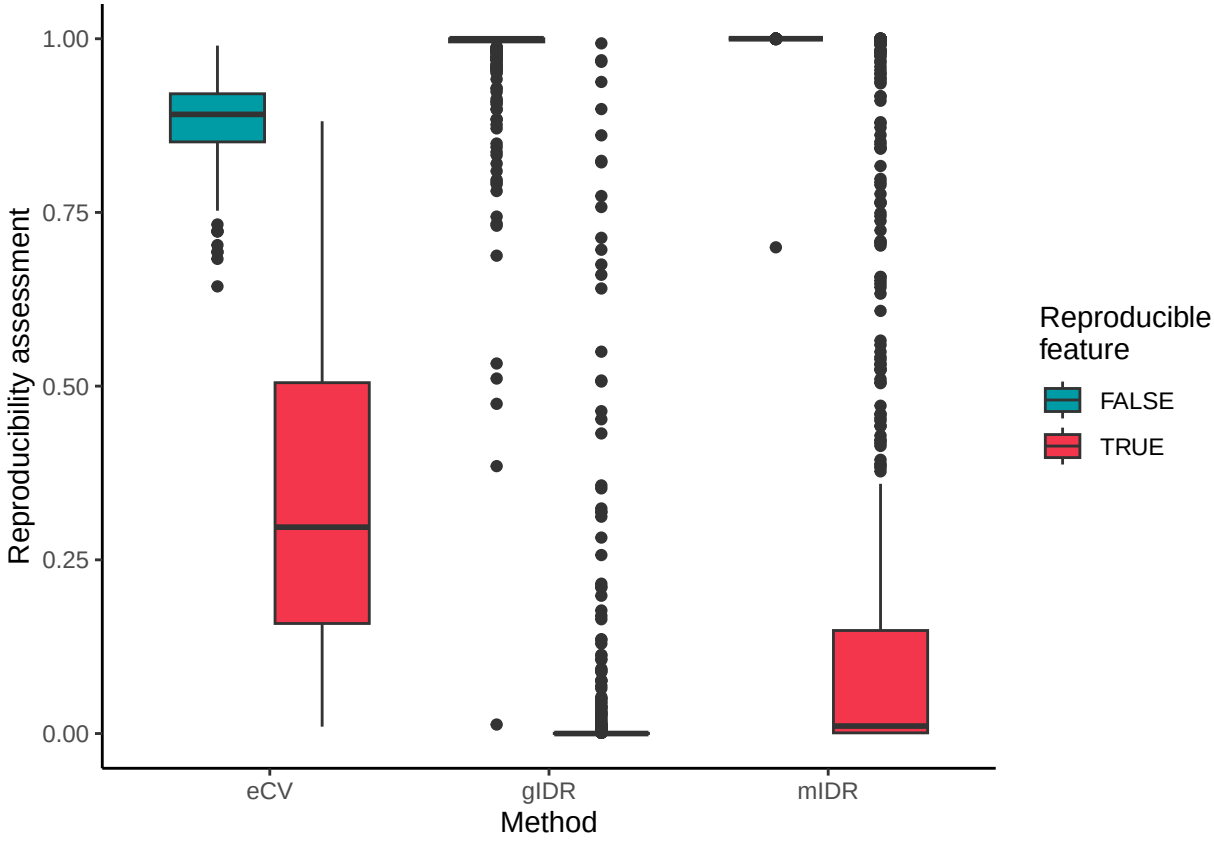
Here is an example where we call function `mrep_assessment` interactively on the previous simulated data, using multiple methods.

```
# Define parameters for each method.
params <- list(
  eCV = list(max.ite = 100),
  gIDR = list(
    mu = 2,
    sigma = 1.3,
    rho = 0.8,
    p = 0.7,
    eps = 1e-3,
    max.ite = 50
  ),
  mIDR = list(
    mu = 2,
    sigma = 1.3,
    rho = 0.8,
    p = 0.7,
    eps = 1e-3,
    max.ite = 50
  )
)

# Create a list to store results
results <- NULL
# Loop through methods and calculate reproducibility
for (method in c("eCV", "gIDR", "mIDR")) {
```

```
  results <- results %>%
    bind_rows(data.frame(
      value =
        mrep_assessment(
          x = out$sim_data,
          method = method,
          param = params[[method]]
        )$rep_index,
      Method = method,
      group = out$sim_params$feature_group
    ))
}

# Plot results
 results %>%
 mutate(group = ifelse(group == 1,"FALSE","TRUE")) %>%
 ggplot(aes(x=Method, y = value,fill=group)) +
 scale_fill_manual(values = c( "#009CA6" , "#F4364C")) +
 geom_boxplot() +
 theme_classic() +
 labs(y="Reproducibility assessment", fill="Reproducible\nfeature")
```



References

Li, Qunhua, James B. Brown, Haiyan Huang, and Peter J. Bickel. 2011. "Measuring Reproducibility of High-Throughput Experiments." *The Annals of Applied Statistics* 5 (3). https://doi.org/10.1214/11-AOAS466.