Package ‘evaluomeR’

May 9, 2024

Type Package
Title Evaluation of Bioinformatics Metrics
URL https://github.com/neobernad/evaluomeR
Version 1.20.0
BugReports https://github.com/neobernad/evaluomeR/issues
Description Evaluating the reliability of your own metrics and the measurements done on your own datasets by analysing the stability and goodness of the classifications of such metrics.
License GPL-3
Encoding UTF-8
LazyData true
Depends R (>= 3.6), SummarizedExperiment, MultiAssayExperiment, cluster (>= 2.0.9), fpc (>= 2.2-3), randomForest (>= 4.6.14), flexmix (>= 2.3.15)
Imports corrplot (>= 0.84), grDevices, graphics, reshape2, ggplot2, ggdendro, plotrix, stats, matrixStats, Rdpack, MASS, class, prabclus, mclust, kableExtra
Suggests BiocStyle, knitr, rmarkdown, magrittr
VignetteBuilder knitr
RdMacros Rdpack
RoxygenNote 7.0.2
biocViews Clustering, Classification, FeatureExtraction
Collate 'data.R' 'helpers.R' 'internalClusterboot.R' 'internalFunctions.R' 'stabilityIndex.R' 'qualityIndices.R' 'correlation.R' 'metricsAnalysis.R' 'predictions.R'
git_url https://git.bioconductor.org/packages/evaluomeR
git_branch RELEASE_3_19
git_last_commit 02887fc
bioMetrics

Dataset: Metrics for biological pathways

Description

Metrics for biological pathways, 2 metrics that quantitative characterizations of the importance of regulation in biochemical pathway systems, including systems designed for applications in synthetic biology or metabolic engineering. The metrics are reachability and efficiency.
Usage

data("bioMetrics")

Format

An object of class `SummarizedExperiment` with 15 rows and 3 columns.

References


evaluomeRSupportedCBI  
*Get supported CBIs in evaluomeR.*

description

A vector of supported CBIs available in evaluomeR.

Usage

evaluomeRSupportedCBI()

Value

A String vector.

Examples

supportedCBIs <- evaluomeRSupportedCBI

gDataQualityRange  
*Dataframe getter for qualityRange function.*

description

This method is a wrapper to retrieve a specific `SummarizedExperiment` given a k value from the object returned by *qualityRange* function.

Usage

gDataQualityRange(data, k)
Arguments

- **data**: The object returned by `qualityRange` function.
- **k**: The desired k cluster.

Value

The `SummarizedExperiment` that contains information about the selected k cluster.

Examples

```r
# Using example data from our package
data("ontMetrics")
qualityRangeData <- qualityRange(ontMetrics, k.range=c(3,5), getImages = FALSE)
# Getting dataframe that contains information about k=5
k5Data = getDataQualityRange(qualityRangeData, 5)
```

---

**getOptimalKValue**

*Calculating the optimal value of k.*

Description

This method finds the optimal value of K per each metric.

Usage

```r
getOptimalKValue(stabData, qualData, k.range = NULL)
```

Arguments

- **stabData**: An output `ExperimentList` from a `stabilityRange` execution.
- **qualData**: An output `SummarizedExperiment` from a `qualityRange` execution.
- **k.range**: A range of K values to limit the scope of the analysis.

Value

It returns a dataframe following the schema: metric, optimal_k.

Examples

```r
# Using example data from our package
data("rnaMetrics")
stabilityData <- stabilityRange(data=rnaMetrics, k.range=c(2,4), bs=20, getImages = FALSE)
qualityData <- qualityRange(data=rnaMetrics, k.range=c(2,4), getImages = FALSE)
kOptTable = getOptimalKValue(stabilityData, qualityData)
```
globalMetric

Global metric score defined by a prediction.

Description
This analysis calculates a global metric score based upon a prediction model computed with flexmix package.

Usage

```r
globalMetric(data, k.range = c(2, 15), nrep = 10,
    criterion = c("BIC", "AIC"), PCA = FALSE, seed = NULL)
```

Arguments

- **data**
  A SummarizedExperiment. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.

- **k.range**
  Concatenation of two positive integers. The first value k.range[1] is considered as the lower bound of the range, whilst the second one, k.range[2], as the higher. Both values must be contained in [2,15] range.

- **nrep**
  Positive integer. Number of random initializations used in adjusting the model.

- **criterion**
  String. Criterion applied in order to select the best model. Possible values: "BIC" or "AIC".

- **PCA**
  Boolean. If true, a PCA is performed on the input dataframe before computing the predictions.

- **seed**
  Positive integer. A seed for internal bootstrap.

Value

A dataframe containing the global metric score for each metric.

Examples

```r
# Using example data from our package
data("rnaMetrics")
globalMetric(rnaMetrics, k.range = c(2,3), nrep=10, criterion="AIC", PCA=TRUE)
```
metricsCorrelations  *Calculation of Pearson correlation coefficient.*

Description

Calculation of Pearson correlation coefficient between every pair of metrics available in order to quantify their interrelationship degree. The score is in the range [-1,1]. Perfect correlations: -1 (inverse), and 1 (direct).

Usage

```
metricsCorrelations(data, margins = c(0, 10, 9, 11), getImages = TRUE)
```

Arguments

- **data**: A `SummarizedExperiment`. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.
- **margins**: See `par`.
- **getImages**: Boolean. If true, a plot is displayed.

Value

The Pearson correlation matrix as an assay in a `SummarizedExperiment` object.

Examples

```
# Using example data from our package
data("ontMetrics")
cor = metricsCorrelations(ontMetrics, getImages = TRUE, margins = c(1,0,5,11))
```

ontMetrics  *Dataset: Structural ontology metrics*

Description

Structural ontology metrics, 19 metrics measuring structural aspects of bio-ontologies have been analysed on two different corpora of ontologies: OBO Foundry and AgroPortal

Usage

```
data("ontMetrics")
```
Format
  An object of class SummarizedExperiment with 80 rows and 20 columns.

References

Description
  It plots the value of the metrics in a SummarizedExperiment object as a boxplot.

Usage
  plotMetricsBoxplot(data)

Arguments
  data A SummarizedExperiment. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.

Value
  Nothing.

Examples
  # Using example data from our package
data("ontMetrics")
plotMetricsBoxplot(ontMetrics)
plotMetricsCluster  

Metric values clustering.

Description

It clusters the value of the metrics in a **SummarizedExperiment** object as a boxplot.

Usage

```r
plotMetricsCluster(data, scale = FALSE)
```

Arguments

- `data`: A **SummarizedExperiment**. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.
- `scale`: Boolean. If true input data is scaled. Default: FALSE.

Value

An hclust object.

Examples

```r
# Using example data from our package
data("ontMetrics")
plotMetricsCluster(ontMetrics, scale=TRUE)
```

plotMetricsClusterComparison  

Comparison between two clusterings as plot. plotMetricsClusterComparison

Description

It plots a clustering comparison between two different k-cluster vectors for a set of metrics.

Usage

```r
plotMetricsClusterComparison(data, k.vector1, k.vector2, seed = NULL)
```
plotMetricsMinMax

Arguments

- **data**: A `SummarizedExperiment`. The `SummarizedExperiment` must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contain the names of the metrics. The rows contain the measurements of the metrics for each instance in the dataset.

- **k.vector1**: Vector of positive integers representing `k` clusters. The `k` values must be contained in [2,15] range.

- **k.vector2**: Vector of positive integers representing `k` clusters. The `k` values must be contained in [2,15] range.

- **seed**: Positive integer. A seed for internal bootstrap.

Value

- Nothing.

Examples

```r
# Using example data from our package
data("rnaMetrics")
stabilityData <- stabilityRange(data=rnaMetrics, k.range=c(2,4), bs=20, getImages = FALSE)
qualityData <- qualityRange(data=rnaMetrics, k.range=c(2,4), getImages = FALSE)
kOptTable = getOptimalKValue(stabilityData, qualityData)
```

Description

It plots the minimum, maximum and standard deviation values of the metrics in a `SummarizedExperiment` object.

Usage

```r
plotMetricsMinMax(data)
```

Arguments

- **data**: A `SummarizedExperiment`. The `SummarizedExperiment` must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contain the names of the metrics. The rows contain the measurements of the metrics for each instance in the dataset.
plotMetricsViolin

**Value**

Nothing.

**Examples**

```r
# Using example data from our package
data("ontMetrics")
plotMetricsMinMax(ontMetrics)
```

---

**plotMetricsViolin**

*Metric values as violin plot.*

**Description**

It plots the value of the metrics in a `SummarizedExperiment` object as a violin plot.

**Usage**

```r
plotMetricsViolin(data)
```

**Arguments**

- `data` A `SummarizedExperiment`. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contain the names of the metrics. The rows contain the measurements of the metrics for each instance in the dataset.

**Value**

Nothing.

**Examples**

```r
# Using example data from our package
data("ontMetrics")
plotMetricsViolin(ontMetrics)
```
The goodness of the classifications are assessed by validating the clusters generated. For this purpose, we use the Silhouette width as validity index. This index computes and compares the quality of the clustering outputs found by the different metrics, thus enabling to measure the goodness of the classification for both instances and metrics. More precisely, this goodness measurement provides an assessment of how similar an instance is to other instances from the same cluster and dissimilar to all the other clusters. The average on all the instances quantifies how appropriately the instances are clustered. Kaufman and Rousseeuw suggested the interpretation of the global Silhouette width score as the effectiveness of the clustering structure. The values are in the range \([0,1]\), having the following meaning:

- There is no substantial clustering structure: \([-1, 0.25]\).
- The clustering structure is weak and could be artificial: \([0.25, 0.50]\).
- There is a reasonable clustering structure: \([0.50, 0.70]\).
- A strong clustering structure has been found: \([0.70, 1]\).

**Usage**

```r
def quality(data, k = 5, cbi = "kmeans", getImages = TRUE, seed = NULL)
```

**Arguments**

- **data**: A `SummarizedExperiment`. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.

- **k**: Positive integer. Number of clusters between \([2,15]\) range.

- **cbi**: Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam", "hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with '_pam' makes use of pam. The method used in 'hclust' CBI is "ward.D2".

- **getImages**: Boolean. If true, a plot is displayed.

- **seed**: Positive integer. A seed for internal bootstrap.

**Value**

A `SummarizedExperiment` containing the silhouette width measurements and cluster sizes for cluster \(k\).
References


Examples

```r
# Using example data from our package
data("ontMetrics")
result = quality(ontMetrics, k=4)
```

**qualityRange**

*Goodness of classifications for a range of k clusters.*

**Description**

The goodness of the classifications are assessed by validating the clusters generated for a range of k values. For this purpose, we use the Silhouette width as validity index. This index computes and compares the quality of the clustering outputs found by the different metrics, thus enabling to measure the goodness of the classification for both instances and metrics. More precisely, this measurement provides an assessment of how similar an instance is to other instances from the same cluster and dissimilar to the rest of clusters. The average on all the instances quantifies how the instances appropriately are clustered. Kaufman and Rousseeuw suggested the interpretation of the global Silhouette width score as the effectiveness of the clustering structure. The values are in the range [0,1], having the following meaning:

- There is no substantial clustering structure: [-1, 0.25].
- The clustering structure is weak and could be artificial: ]0.25, 0.50[.
- There is a reasonable clustering structure: ]0.50, 0.70[.
- A strong clustering structure has been found: ]0.70, 1[.

**Usage**

```r
qualityRange(data, k.range = c(3, 5), cbi = "kmeans",
getImages = TRUE, seed = NULL)
```

**Arguments**

- **data**
  A *SummarizedExperiment*. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.

- **k.range**
  Concatenation of two positive integers. The first value `k.range[1]` is considered as the lower bound of the range, whilst the second one, `k.range[2]`, as the higher. Both values must be contained in [2,15] range.
cbi Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam", "hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with "_pam" makes use of pam. The method used in 'hclust' CBI is "ward.D2".

getImages Boolean. If true, a plot is displayed.

seed Positive integer. A seed for internal bootstrap.

Value

A list of SummarizedExperiment containing the silhouette width measurements and cluster sizes from k.range[1] to k.range[2]. The position on the list matches with the k-value used in that dataframe. For instance, position 5 represents the dataframe with k = 5.

References


Examples

# Using example data from our package
data("ontMetrics")
# Without plotting
dataFrameList = qualityRange(ontMetrics, k.range=c(2,3), getImages = FALSE)

qualitySet Goodness of classifications for a set of k clusters.

Description

The goodness of the classifications are assessed by validating the clusters generated for a range of k values. For this purpose, we use the Silhouette width as validity index. This index computes and compares the quality of the clustering outputs found by the different metrics, thus enabling to measure the goodness of the classification for both instances and metrics. More precisely, this measurement provides an assessment of how similar an instance is to other instances from the same cluster and dissimilar to the rest of clusters. The average on all the instances quantifies how the instances appropriately are clustered. Kaufman and Rousseeuw suggested the interpretation of the global Silhouette width score as the effectiveness of the clustering structure. The values are in the range [0,1], having the following meaning:

- There is no substantial clustering structure: [-1, 0.25].
- The clustering structure is weak and could be artificial: [0.25, 0.50].
- There is a reasonable clustering structure: [0.50, 0.70].
- A strong clustering structure has been found: [0.70, 1].
Usage

```r
qualitySet(data, k.set = c(2, 4), cbi = "kmeans", getImages = TRUE,
            seed = NULL)
```

Arguments

- `data` A `SummarizedExperiment`. The `SummarizedExperiment` must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contain the names of the metrics. The rows contain the measurements of the metrics for each instance in the dataset.
- `k.set` A list of integer values of k, as in c(2,4,8). The values must be contained in [2,15] range.
- `cbi` Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam", "hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with '_pam' makes use of `pam`. The method used in 'hclust' CBI is "ward.D2".
- `getImages` Boolean. If true, a plot is displayed.
- `seed` Positive integer. A seed for internal bootstrap.

Value

A list of `SummarizedExperiment` containing the silhouette width measurements and cluster sizes from `k.set`.

References


Examples

```r
# Using example data from our package
data("rnaMetrics")
# Without plotting
dataFrameList = qualitySet(rnaMetrics, k.set=c(2,3), getImages = FALSE)
```

**Description**

RNA quality metrics for the assessment of gene expression differences, 2 quality metrics from 16 aliquots of a unique batch of RNA Samples. The metrics are Degradation Factor (DegFact) and RNA Integrity Number (RIN)
### st.

#### Usage

```r
data("rnaMetrics")
```

#### Format

An object of class `SummarizedExperiment` with 16 rows and 3 columns.

#### References


---

### Stability

#### Description

This analysis permits to estimate whether the clustering is meaningfully affected by small variations in the sample. First, a clustering using the k-means algorithm is carried out. The value of `k` can be provided by the user. Then, the stability index is the mean of the Jaccard coefficient values of a number of `bs` bootstrap replicates. The values are in the range [0,1], having the following meaning:

- Unstable: [0, 0.60[.
- Doubtful: [0.60, 0.75[.
- Stable: ]0.75, 0.85[.
- Highly Stable: ]0.85, 1[.

#### Usage

```r
stability(data, k = 5, bs = 100, cbi = "kmeans", getImages = TRUE, seed = NULL)
```

#### Arguments

- **data**: A `SummarizedExperiment`. The `SummarizedExperiment` must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.
- **k**: Positive integer. Number of clusters between [2,15] range.
- **bs**: Positive integer. Bootstrap value to perform the resampling.
- **cbi**: Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam", "hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with '_pam' makes use of `pam`. The method used in 'hclust' CBI is "ward.D2".
- **getImages**: Boolean. If true, a plot is displayed.
- **seed**: Positive integer. A seed for internal bootstrap.
stabilityRange

Value

A ExperimentList containing the stability and cluster measurements for k clusters.

References


Examples

# Using example data from our package
data("ontMetrics")
result <- stability(ontMetrics, k=6, getImages=TRUE)

# stabilityRange function

Description

This analysis permits to estimate whether the clustering is meaningfully affected by small variations in the sample. For a range of k values (k.range), a clustering using the k-means algorithm is carried out. Then, the stability index is the mean of the Jaccard coefficient values of a number of bs bootstrap replicates. The values are in the range [0,1], having the following meaning:

- Unstable: [0, 0.60[.
- Doubtful: [0.60, 0.75[.
- Stable: ]0.75, 0.85[.
- Highly Stable: ]0.85, 1[.

Usage

stabilityRange(data, k.range = c(2, 15), bs = 100, cbi = "kmeans",
getImages = TRUE, seed = NULL)

Arguments

data A SummarizedExperiment. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.
stabilitySet

k.range  Concatenation of two positive integers. The first value k.range[1] is con-
idered as the lower bound of the range, whilst the second one, k.range[2], as the
higher. Both values must be contained in [2,15] range.

bs  Positive integer. Bootstrap value to perform the resampling.

cbi  Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam",
"hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with '_pam' makes
use of pam. The method used in 'hclust' CBI is "ward.D2".

getImages  Boolean. If true, a plot is displayed.

seed  Positive integer. A seed for internal bootstrap.

Value

A ExperimentList containing the stability and cluster measurements for 2 to k clusters.

References


Jaccard P (1901). “Distribution de la flore alpine dans le bassin des Dranses et dans quelques

Examples

# Using example data from our package
data("ontMetrics")
result <- stabilityRange(ontMetrics, k.range=c(2,3))

---

stabilitySet  Stability index for a set of k clusters.

Description

This analysis permits to estimate whether the clustering is meaningfully affected by small variations
in the sample. For a set of k values (k.set), a clustering using the k-means algorithm is carried out.
Then, the stability index is the mean of the Jaccard coefficient values of a number of bs bootstrap
replicates. The values are in the range [0,1], having the following meaning:

- Unstable: [0, 0.60).
- Doubtful: [0.60, 0.75].
- Stable: [0.75, 0.85].
- Highly Stable: ]0.85, 1].

Usage

stabilitySet(data, k.set = c(2, 3), bs = 100, cbi = "kmeans",
getImages = TRUE, seed = NULL)
stabilitySet

Arguments

data A SummarizedExperiment. The SummarizedExperiment must contain an assay with the following structure: A valid header with names. The first column of the header is the ID or name of the instance of the dataset (e.g., ontology, pathway, etc.) on which the metrics are measured. The other columns of the header contains the names of the metrics. The rows contains the measurements of the metrics for each instance in the dataset.
k.set A list of integer values of k, as in c(2,4,8). The values must be contained in [2,15] range.
bs Positive integer. Bootstrap value to perform the resampling.
cbi Clusterboot interface name (default: "kmeans"): "kmeans", "clara", "clara_pam", "hclust", "pamk", "pamk_pam", "pamk". Any CBI appended with '_pam' makes use of pam. The method used in 'hclust' CBI is "ward.D2".
getImages Boolean. If true, a plot is displayed.
seed Positive integer. A seed for internal bootstrap.

Value

A ExperimentList containing the stability and cluster measurements of the list of k clusters.

References


Examples

# Using example data from our package
data("rnaMetrics")
result <- stabilitySet(rnaMetrics, k.set=c(2,3))
Index

* datasets
  bioMetrics, 2
  ontMetrics, 6
  rnaMetrics, 14

bioMetrics, 2

evaluomeRSupportedCBI, 3
ExperimentList, 4, 16–18

flexmix, 5

dataQualityRange, 3
getOptimalKValue, 4

metricsCorrelations, 6

ontMetrics, 6

pam, 11, 13–15, 17, 18
par, 6

plotMetricsBoxplot, 7
plotMetricsCluster, 8
plotMetricsClusterComparison, 8
plotMetricsMinMax, 9
plotMetricsViolin, 10

quality, 11

qualityRange, 3, 4, 12

qualitySet, 13

rnaMetrics, 14

stability, 15

stabilityRange, 4, 16

stabilitySet, 17

SummarizedExperiment, 3–16, 18