Package ‘COMMUNAL’

February 19, 2015

Type Package
Title Robust Selection of Cluster Number K
Version 1.0
Date 2014-02-11
Author Albert Chen [aut, cre], Timothy E Sweeney [aut], Olivier Gevaert [ths]
Maintainer Albert Chen <acc2015@stanford.edu>
Depends R (>= 2.10), cluster
Suggests RUnit, NMF, ConsensusClusterPlus
Imports clValid, fpc, methods, rgl
Description Facilitates optimal clustering of a data set. Provides a framework to run a wide range of clustering algorithms to determine the optimal number (k) of clusters in the data. Then analyzes the cluster assignments from each clustering algorithm to identify samples that repeatedly classify to the same group. We call these ‘core clusters’, providing a basis for later class discovery.
NeedsCompilation no
License GPL-2
Repository CRAN
Date/Publication 2015-02-11 19:55:20

R topics documented:

COMMUNAL-package .................................................. 2
BRCA.100 ................................................................. 3
BRCA.results .............................................................. 4
clusterKeys ............................................................... 4
clusterRange .............................................................. 5
COMMUNAL ................................................................. 6
COMMUNAL-class .......................................................... 9
examineCounts ........................................................... 10
plotRange3D ............................................................... 11
returnCore ............................................................... 13

Index 15
Description

This package allows for identification of optimal clustering for a data set. It provides a framework to run a wide range of clustering algorithms to determine the optimal number (k) of clusters in the data. It then provides a function to analyze the cluster assignments from each clustering algorithm to identify samples that repeatedly classify to the same group. We call these 'core clusters,' leading to optimal beds for later class discovery.

Details

Package: COMMUNAL
Type: Package
Version: 1.0
Date: 2015-01-05
License: GPL-2
Imports: clValid, fpc, methods
Depends: R (>= 2.10), cluster
Suggests: RUnit, NMF, ConsensusClusterPlus, rgl

Start with a matrix of data to cluster. Important functions are:
- communal to run clustering algorithms
- clusterrange to run clustering algorithms (harness for COMMUNAL)
- plotRange3D to pick k
- clusterkeys to identify core clusters
- returnCore to identify core clusters

Author(s)

Albert Chen, Timothy E Sweeney, Olivier Gevaert
Maintainer: Albert Chen acc2015@stanford.edu

Examples

```r
# Not run:
# create artificial data set with 3 distinct clusters
set.seed(1)
V1 = c(abs(rnorm(100, 2)), abs(rnorm(100, 50)), abs(rnorm(100, 140)))
V2 = c(abs(rnorm(100, 2, 8)), abs(rnorm(100, 55, 4)), abs(rnorm(100, 105, 1)))
data <- t(data.frame(V1, V2))
colnames(data) <- paste("Sample", 1:ncol(data), sep="")
rownames(data) <- paste("Gene", 1:nrow(data), sep="")
```
```r
## run COMMUNAL
result <- COMMUNAL(data=data, ks=seq(2,5))  # result is a COMMUNAL object
k <- 3                                      # suppose optimal cluster number is 3
clusters <- result$getClustering(k)        # method to extract clusters
mat.key <- clusterKeys(clusters, k=k)      # get core clusters
examineCounts(mat.key)                    # help decide agreement.thresh
core <- returnCore(mat.key, agreement.thresh=50)  # find 'core' clusters (all algs agree)
table(core)  # the 'core' clusters

## Additional arguments are passed down to clValid, NMF, ConsensusClusterPlus
result <- COMMUNAL(data=data, ks=2:5, clus.methods=c("diana", "ccp-hc", "nmf"), reps=20, nruns=2)

## To identify k, use clusterRange and plotRange3D to visualize validation measures
data(BRCA.100)  # 533 tissues to cluster, with measurements of 100 genes each
varRange <- c(50, 75, 100)
clus.methods <- c("hierarchical", "kmeans")
validation <- c('wb.ratio', 'dunn', 'avg.silwidth')
range.results <- clusterRange(BRCA.100, varRange, ks=2:5, clus.methods=clus.methods, validation=validation)
plot.data <- plotRange3D(range.results, ks=2:5, clus.methods, validation)

## End(Not run)
```

---

**BRCA.100**

*Breast Cancer gene expression data.*

---

**Description**

Expression data of 100 genes (rows) in 533 conditions (columns).

**Usage**

```r
data("BRCA.100")
```

**Format**

A data frame with 533 observations on 100 variables.

**Source**


**Examples**

```r
data(BRCA.100)
```
BRCA.results  |  Breast Cancer clusterRange output

**Description**

Output of `clusterRange` on breast cancer gene expression dataset. To be displayed with `plotRange3D` for visualization.

**Usage**

```r
data("BRCA.results")
```

**Format**

List of 2. First element is a list of 11 `COMMUNAL` objects. Second element is a vector of the corresponding number of data points that were clustered.

**Examples**

```r
data(BRCA.results)
```

---

**clusterKeys**

Rekey cluster assignments.

**Description**

Reindexes (rekeys) the cluster assignments to maximize overlap across algorithms. Ignores algorithms which could not find k clusters; i.e. when one of the clusters is smaller than the min.size argument. Use this after determining the number of clusters.

**Usage**

```r
clusterKeys(clusters, k, min.size = 3)
```

**Arguments**

- `clusters`: Data frame of cluster assignments, where rows are samples, columns are algorithms, assignments are integers. For example, the output of the `getClustering` method in "COMMUNAL".
- `k`: Number of clusters selected.
- `min.size`: Minimum cluster size. Algorithms that return clusters smaller than this (or that don’t have k clusters) are tossed out.

---
Value

Returns a matrix of rekeyed cluster assignments, such that cluster 'n' refers to the same cluster across all algorithms. Cluster 0 contains the samples for which no consistent 'core' cluster could be identified.

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

Examples

# reindexes cluster numbers to agree
k <- 3
custers <- data.frame(
    alg1=as.integer(c(1,1,1,1,2,2,2,2,2,3,3,3,3)),
    alg2=as.integer(c(1,1,1,1,3,3,3,3,3,2,2,2,2)),
    alg3=as.integer(c(3,3,3,3,3,1,1,1,1,2,2,2,2))
)
mat.key <- clusterKeys(clusters, k)
mat.key # cluster indices are relabeled
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters
table(core) # the 'core' clusters

# some clusters assignments are undetermined
k <- 3
custers <- data.frame(
    alg1=as.integer(c(1,1,1,1,2,2,2,2,2,3,3,3,3,3,1,1,2,2,3,3)),
    alg2=as.integer(c(1,1,1,1,2,2,2,2,2,3,3,3,3,3,1,2,2,3,3,1)),
    alg3=as.integer(c(1,1,1,1,2,2,2,2,2,3,3,3,3,3,2,3,1,1,2,3))
)
mat.key <- clusterKeys(clusters, k)
mat.key # last six samples have conflicting assignments
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=66) # at least 2 of 3 algs must agree
table(core)
core <- returnCore(mat.key, agreement.thresh=99) # all algs must agree
table(core)

clusterRange

Description

Convenience harness to run COMMUNAL on a range of inputs, for a fixed set of parameters. The columns are the points to be clustered. The goal is to reduce the dimensionality of the data points, and try clustering with a varying number of dimensions. The rows are (optionally) first sorted by variance. The top x rows are used for clustering by COMMUNAL, for each x in varRange. Output is used by plotRange3D to generate 3D plot.
Usage

\texttt{clusterRange(datumtx, varRange, \ldots)}

Arguments

\begin{itemize}
  \item \texttt{datumtx} \quad The data for input to \texttt{COMMUNAL}.
  \item \texttt{varRange} \quad Numeric vector of how many items of data matrix to cluster. \texttt{clusterRange} runs \texttt{COMMUNAL} on the \texttt{x} rows (dimensions) with the highest variance for each element \texttt{x} in \texttt{varRange}.
  \item \ldots \quad Arguments to pass down to \texttt{COMMUNAL}. If \texttt{reorder=FALSE} is specified, then the rows will not be first sorted by variance in this function, and the rows will be taken in the given order. By default, rows are sorted by variance.
\end{itemize}

Value

\begin{itemize}
  \item \texttt{all.results} \quad list of \texttt{COMMUNAL} objects, one for each range in \texttt{varRange}
  \item \texttt{varRange} \quad the \texttt{varRange} input parameter
\end{itemize}

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

Examples

```r
## Not run:
## To identify k, use clusterRange and plotRange3D to visualize validation measures
data(BRCA.100) # 533 tissues to cluster, with measurements of 100 genes each
varRange <- c(50, 75, 100)
clus.methods <- c("hierarchical", "kmeans")
validation <- c("wb.ratio", "dunn", "avg.silwidth")
range.results <- clusterRange(BRCA.100, varRange, ks=2:5, clus.methods=clus.methods,
                               validation=validation)
plot.data <- plotRange3D(range.results, ks=2:5, clus.methods, validation)
## Note: the BRCA.results dataset was generated by running clusterRange on
## a larger range than the one here (with a larger input dataset)

## End(Not run)
```

\textbf{COMMUNAL} \quad \textit{Run clustering algorithms and evaluate validation metrics.}
Description

This function runs various (user-specified) clustering algorithms on the data, for each potential number of clusters \( k \). It then runs internal validation measures to quantify the fit of each clustering. The returned object is of class "COMMUNAL", and can be used to identify 'core' clusters in the data. Currently supported clustering algorithms are those in packages "clValid", "NMF", and "ConsensusClusterPlus".

To determine the optimal number of clusters, use the `clusterRange` and `plotRange3D` functions.

Usage

```r
COMMUNAL(data, ks = 2:10, clusNmethods = c("hierarchical", "kmeans"),
          validation = c("Connectivity", "dunn", "wb.ratio", "g3", "g2",
                       "pearsongamma", "avg.silwidth", "sindex"),
          dist.metric = "euclidean", agglNmethod = "average",
          neighb.size = 10, seed = NULL, ...)
```

Arguments

- **data** The data to cluster (numeric matrix or data frame). The columns are clustered, rows are features. If using cluster method `nmf`, all entries must be non-negative.
- **ks** A numeric vector of integers greater than 1, for the number of clusters to consider. For example, 2:4 tells the function to try clusterings with 2, 3, and 4 clusters.
- **validation** A character vector of the validation measures to consider. Valid options: "Connectivity", "average.between", "g2", "ch", "sindex", "avg.silwidth", "average.within", "dunn", "widestgap", "wb.ratio", "entropy", "dunn2", "pearsongamma", "g3", "within.cluster.ss", "min.separation", "max.diameter". With the exception of "Connectivity", which is calculated by clValid::connectivity, these are calculated with fpc::cluster.stats.
- **dist.metric** Which metric to use when calculating the distance matrix. Used by clValid clustering algorithms, and in calculating validation measures. Available choices are "euclidean", "correlation", "manhattan".
- **agglNmethod** The agglomeration method to use for "hclust" and "agnes" (if specified in clusNmethods). Available choices are "ward", "single", "complete", "average".
- **neighb.size** Numeric value. The neighborhood size used for calculating the Connectivity validation measure.
- **seed** Numeric value. Random seed to use in ConsensusClusterPlus and NMF.
- **...** Other arguments to pass down to ConsensusClusterPlus, NMF, and clValid.
Value

Return object is an object of class \texttt{COMMUNAL}. The class has a \texttt{getClustering} method to extract a data frame of cluster assignments. Alternatively, functions \texttt{clusterKeys} and \texttt{returnCore} are provided to identify core clusters. See examples below.

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

See Also

Class \texttt{"COMMUNAL"}. Use functions \texttt{clusterKeys} and \texttt{returnCore} to identify core clusters.

Examples

```r
## Not run:
## create artificial data set with S distinct clusters
set.seed(1)
V1 = c(abs(rnorm(100, 2)), abs(rnorm(100, 50)), abs(rnorm(100, 140)))
V2 = c(abs(rnorm(100, 2, 8)), abs(rnorm(100, 55, 4)), abs(rnorm(100, 105, 1)))
data <- t(data.frame(V1, V2))
colnames(data) <- paste("Sample", 1:ncol(data), sep="")
rownames(data) <- paste("Gene", 1:nrow(data), sep="")

## run COMMUNAL
result <- COMMUNAL(data=data, ks=seq(3,5)) # result is a COMMUNAL object
k <- 3 # suppose optimal cluster number is 3
clusters <- result$getClustering(k) # method to extract clusters
mat.key <- clusterKeys(clusters, k=k) # get core clusters
examineCounts(mat.key) # help decide agreement.thresh
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters (all alg agree)
table(core) # the 'core' cluster sizes
## Note: could try a different value for k to
## see clusters with sub-optimal k

## Can specify clustering methods and validation measures
result <- COMMUNAL(data=data, ks=c(2,3),
    clus.methods = c("diana", "som", "pam", "kmeans", "ccp-hc", "nmf"),
    validation=c("pearsongamma", 'avg.silwidth'))
clusters <- result$getClustering(k=3)
mat.key <- clusterKeys(clusters, k=3)
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters
table(core) # the 'core' clusters

## Additional arguments are passed down to clValid, NMF, ConsensusClusterPlus
result <- COMMUNAL(data=data, ks=2:5,
    clus.methods=c("diana", "ccp-hc", "nmf"), reps=20, nruns=2)
## End(Not run)
```
Description

This is the class of the object returned by the `COMMUNAL` function. It contains all the clustering results, including the values of the validation measures, and the cluster assignments. Cluster assignments for a particular number of clusters can be conveniently extracted with the `getClustering(k)` method.

Extends

All reference classes extend and inherit methods from "envRefClass".

Fields

- `cluster.list`: Object of class list List of all cluster assignments from each algorithm. Use the `getClustering` method to conveniently extract clusters for a given value of k.
- `measures`: Object of class array The validation measures calculated
- `clus.methods`: Object of class character The clustering algorithms used.
- `ks`: Object of class numeric The range of cluster numbers tested.
- `validation`: Object of class character The validation measures used.
- `dist.metric`: Object of class character The distance metric used to calculate validation scores.
- `item.names`: Object of class character The names of the clustered items.
- `call`: Object of class call The call to `COMMUNAL` used to create the object.

Methods

- `getClustering(k)`: For given value of k, extract the cluster assignments from each clustering algorithm.
- `show()`: Default print method.

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

See Also

Vignette and `COMMUNAL`. The function `COMMUNAL` returns an object of this class.
Examples

showClass("COMMUNAL")
## Not run:
## create artificial data set with 3 distinct clusters
set.seed(1)
V1 = c(abs(rnorm(100, 2)), abs(rnorm(100, 50)), abs(rnorm(100, 140)))
V2 = c(abs(rnorm(100, 2, 8)), abs(rnorm(100, 55, 4)), abs(rnorm(100, 105, 1)))
data <- t(data.frame(V1, V2))
colnames(data) <- paste("Sample", 1:ncol(data), sep="")
rownames(data) <- paste("Gene", 1:nrow(data), sep="")

## run COMMUNAL with defaults
result <- COMMUNAL(data=data, ks=seq(2,5)) # result is a COMMUNAL object
k <- 3 # suppose optimal cluster number is 3
clusters <- result$getClustering(k) # extract clusters
mat.key <- clusterKeys(clusters, k=k) # get core clusters
examineCounts(mat.key) # all algorithms agree
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' cluster assignments
table(core) # the 'core' cluster sizes

result$measures # access validation measures

## End(Not run)

examineCounts

Examine algorithm cluster agreement

Description

Returns a data frame to help the user choose the agreement.thresh parameter for returnCore. Each row shows how many samples (sample.count) are at each agreement level (percent.agreement), and what percent of the data would be removed for that agreement threshold (percent.remaining.if.removed).

Usage

examineCounts(mat.key)

Arguments

mat.key A matrix of cluster assignments, where rows are items and columns are algorithms. The output of clusterKeys.

Value

A 3 column table, showing the number of samples at each agreement level and the percent of data above that agreement level. The agreement level for a sample is the (highest) fraction of algorithms which agree on its cluster assignment.
Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

Examples

# reindexes cluster numbers to agree
k <- 3
clusters <- data.frame(
  alg1=as.integer(c(1,1,1,1,1,2,2,2,2,2,3,3,3,3,3)),
  alg2=as.integer(c(1,1,1,1,3,3,3,3,3,2,2,2,2,2)),
  alg3=as.integer(c(3,3,3,3,3,1,1,1,1,2,2,2,2,2))
)
mat.key <- clusterKeys(clusters, k)
mat.key # cluster indices are relabeled
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters
  table(core) # the 'core' clusters

# some clusters assignments are undetermined
k <- 3
clusters <- data.frame(
  alg1=as.integer(c(1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,3,1,1,2,2,3,3)),
  alg2=as.integer(c(1,1,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,3,1,1,2,2,3,3,1)),
  alg3=as.integer(c(1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,3,2,3,1,1,2,3))
)
mat.key <- clusterKeys(clusters, k)
mat.key # last six samples have conflicting assignments
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=66) # need at least 2 of 3 algs to agree
  table(core)
core <- returnCore(mat.key, agreement.thresh=99) # need all algs to agree
  table(core)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)

plotRange3D

Visualize cluster quality

Description

Given COMMUNAL outputs for a dataset, summarize the suitability of a range of cluster numbers according to validation measures. Input can be the output of clusterRange.

Usage

plotRange3D(test_range, ks, goodAlgs, goodMeasures, filename=NULL, ...)
Arguments

test_range list of 2 items, where first element is list of "COMMUNAL" objects, and second is a vector of the number of data points clustered for each element in the list. Can be the output of `clusterRange`.
ks range of cluster number k to plot. Must be subset of ks supplied to `COMMUNAL`.
goodAlgs which algorithms to use in summarizing validation measures. Must be subset of algorithms supplied to `COMMUNAL`.
goodMeasures which validation measures to use in summarizing validation measures. Must be subset of measures supplied to `COMMUNAL`.
filename optionally specify filename to save a snapshot of the 3D image.
... other arguments to pass down to 2D plot of mean z-score against k.

Details

A summarized validation measure value is computed for each value of k, for each dataset. This is done by first subsetting the data to the measures, ks, and algorithms of interest, and then computing averages of the measures for each dataset and k (number of clusters).

For some validation measures, a lower value implies better clustering, and for others a higher value is better. Prior to averaging, measures that favor a lower value are multiplied by negative one. Furthermore, each measure is scaled to have zero mean and unit variance across all the datasets prior to averaging, so each measure has equal weight, and we can compare the plot across datasets.

If the wrong number of clusters was identified

Value

A 3D plot is generated, using the package "rgl". A matrix of the plotted values is returned. A 2D plot of average metric against k is also generated.

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

Examples

```r
## Not run:
data(BRCA.results) # clusterRange output for breast cancer dataset
goodAlgs <- c("hierarchical", "kmeans", "model", "agnes", "som")
goodMeasures <- c("wb.ratio", "avg.silwidth", "dunn")
ks <- 2:8
(values <- plotRange3D(BRCA.results, ks, goodAlgs, goodMeasures))

## End(Not run)
```
returnCore

Assign points to core clusters

Description

Takes keyed cluster assignments of each algorithm and makes final cluster assignments by majority vote. If the majority percentage is \( \leq \) the specified threshold, the cluster cannot be determined and is set to 0.

Usage

\[
\text{returnCore(mat.key, agreement.thresh = 50)}
\]

Arguments

- **mat.key**: Matrix of rekeyed cluster assignments. E.g., the output of `clusterKeys`.
- **agreement.thresh**: Percent of algorithms required to agree for a cluster assignment to be accepted. Otherwise, cluster is set to 0. By default at least half the algorithms must agree.

Details

Can use `examineCounts` to help determine the `agreement.thresh` argument.

Author(s)

Albert Chen and Timothy E Sweeney
Maintainer: Albert Chen acc2015@stanford.edu

Examples

```r
# reindexes cluster numbers to agree
k <- 3
clusters <- data.frame(
  alg1=as.integer(c(1,1,1,1,2,2,2,2,3,3,3,3)),
  alg2=as.integer(c(1,1,1,1,3,3,3,3,2,2,2,2)),
  alg3=as.integer(c(3,3,3,3,3,1,1,1,1,2,2,2))
)
mat.key <- clusterKeys(clusters, k)
mat.key # cluster indices are relabeled
examineCounts(mat.key)
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters
table(core) # the 'core' clusters

# some clusters assignments are undetermined
k <- 3
clusters <- data.frame(
  alg1=as.integer(c(1,1,1,1,2,2,2,2,3,3,3,3,1,1,2,2,2,3,3)),
  alg2=as.integer(c(1,1,1,1,2,2,2,2,3,3,3,3,1,2,2,3,3,1)),
)
alg3 = as.integer(c(1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 3, 1, 1, 2, 3))
)
mat.key <- clusterKeys(clusters, k)
mat.key # last six samples have conflicting assignments
examineCounts(mat.key)
(core <- returnCore(mat.key, agreement.thresh = 66)) # need at least 2 of 3 algs to agree
table(core)
(core <- returnCore(mat.key, agreement.thresh = 99)) # need all algs to agree
table(core)
Index

*Topic **classes**
  COMMUNAL-class, 9
*Topic **cluster**
  COMMUNAL, 6
*Topic **datasets**
  BRCA.100, 3
  BRCA.results, 4
*Topic **package**
  COMMUNAL-package, 2
*Topic **robust**
  COMMUNAL, 6

BRCA.100, 3
BRCA.results, 4

clusterKeys, 2, 4, 8, 10, 13
clusterRange, 2, 4, 5, 7, 12
COMMUNAL, 2, 4–6, 6, 7–9, 11, 12
COMMUNAL-class, 9
COMMUNAL-package, 2

envRefClass, 9
examineCounts, 10, 13

plotRange3D, 2, 4, 5, 7, 11

returnCore, 2, 8, 10, 13