

Package ‘apcluster’

January 2, 2012

Type Package

Title Affinity Propagation Clustering

Version 1.1.1

Date 2011-09-08

Depends R (>= 2.10.1), methods

Imports methods, stats, graphics, utils

Suggests kernlab

Author Ulrich Bodenhofer, Andreas Kothmeier

Maintainer Ulrich Bodenhofer <apcluster@bioinf.jku.at>

Description The apcluster package implements Frey’s and Dueck’s Affinity Propagation clustering in R. The algorithms are largely analogous to the Matlab code published by Frey and Dueck. The package further provides an algorithm for exemplar-based agglomerative clustering that can also be used to join clusters obtained from affinity propagation. Various plotting functions are available for analyzing clustering results.

License GPL (>= 2)

Collate AllClasses.R show-methods.R plot-methods.R labels-methods.R cutree-methods.R aggExCluster.R apcluster.R apclusterLM.R apclusterK.R apclusterDemo.R preferenceRange.R similarity.R sparseToFull.R

URL <http://www.bioinf.jku.at/software/apcluster/>

LazyLoad yes

Repository CRAN

Date/Publication 2011-09-08 09:24:37

R topics documented:

apcluster-package	2
aggExCluster	3
AggExResult-class	6
apcluster	8
apclusterDemo	10
apclusterK	12
APResult-class	13
ch22Promoters	15
cutree-methods	16
ExClust-class	17
labels-methods	19
plot-methods	20
preferenceRange	23
show-methods	25
similarity	26
sparseToFull	28
Index	30

apcluster-package	<i>APCluster Package</i>
-------------------	--------------------------

Description

The apcluster package implements affinity propagation according to Frey and Dueck and a method for exemplar-based agglomerative clustering. It further offers various functions for plotting clustering results.

Details

The central function is `apcluster`. It runs affinity propagation on a given similarity matrix. The function returns an `APResult` object from which the clustering itself and information about the affinity propagation run can be obtained. The package further implements an exemplar-based agglomerative clustering method `aggExCluster` that can be used for computing a complete cluster hierarchy, but also for joining fine-grained clusters previously obtained by affinity propagation clustering. Further functions are implemented to visualize the results and to create distance matrices.

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## run affinity propagation
apres <- apcluster(sim, details=TRUE)

## show details of clustering results
show(apres)

## plot information about clustering run
plot(apres)

## plot clustering result
plot(apres, x)

## employ agglomerative clustering to join clusters
aggres <- aggExCluster(sim, apres)

## show information
show(aggres)
show(cutree(aggres, 2))

## plot dendrogram
plot(aggres)

## plot clustering result for k=2 clusters
plot(aggres, x, k=2)

## plot heatmap
plot(apres, sim)
```

Description

Runs exemplar-based agglomerative clustering for a given similarity matrix

Usage

```
aggExCluster(s, cl)
```

Arguments

`s` an $l \times l$ similarity matrix
`cl` a prior clustering; if present, `cl` must be an object of class `APResult` or `ExClust`.

Details

`aggExCluster` performs agglomerative clustering. Unlike other methods, e.g., the ones implemented in `hclust`, `aggExCluster` is computing exemplars for each cluster and its merging objective is geared towards the identification of meaningful exemplars, too.

For each pair of clusters, the merging objective is computed as follows:

1. An intermediate cluster is created as the union of the two clusters.
2. The potential exemplar is selected from the intermediate cluster as the sample that has the largest average similarity to all other samples in the intermediate cluster.
3. Then the average similarity of the exemplar with all samples in the first cluster and the average similarity with all samples in the second cluster is computed. These two values measure how well the joint exemplar describes the samples in the two clusters.
4. The merging objective is finally computed as the average of the two measures above. Hence, we can consider the merging objective as some kind of “balanced average similarity to the joint exemplar”.

In each step, all pairs of clusters are considered and the pair with the largest merging objective is actually merged. The joint exemplar is then chosen as the exemplar of the merged cluster.

`aggExCluster` can be used in two ways: Firstly, by calling it for a similarity matrix only (with `cl` argument missing). Then a full agglomeration run is performed that starts from 1 clusters (all samples in separate one-element clusters) and ends with one cluster (all samples in one single cluster). Secondly, `aggExCluster` can be called on a similarity matrix along with a prior clustering given by the `cl` argument. In this case, `aggExCluster` starts from the prior clustering. Consequently, a cluster hierarchy with numbers of clusters ranging from the number of clusters in `cl` down to 1 is created.

The result is stored in an `AggExResult` object. If `aggExCluster` was called for a similarity matrix only (argument `cl` missing), `maxNoClusters` is equal to 1, i.e. the result contains a complete cluster hierarchy from 1 clusters (every sample in a separate cluster) to one cluster (every sample in the same cluster). If `aggExCluster` was called for a similarity matrix along with a prior clustering (e.g. obtained from affinity propagation), then agglomeration starts from this clustering, hence, `maxNoClusters` is the number of clusters in the prior clustering. The slot `height` is filled with the merging objective of each of the `maxNoClusters-1` merges. The slot `order` contains a permutation of the samples/clusters for dendrogram plotting. The algorithm for computing this permutation is the same as the one used in `hclust`. If `aggExCluster` was called for a similarity matrix only, the

slot `label` contains the names of the objects to be clustered (if available, otherwise the indices are used). If `aggExCluster` was called for a similarity matrix along with a prior clustering, then labels are set to 'Cluster 1', 'Cluster 2', etc.

Value

Upon successful completion, the function returns an `AggExResult` object.

Author(s)

Ulrich Bodenhofer <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[AggExResult](#), [plot-methods](#), [cutree-methods](#)

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(50,0.2,0.05),rnorm(50,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## compute agglomerative clustering from scratch
aggres1 <- aggExCluster(sim)

## show results
show(aggres1)

## plot dendrogram
plot(aggres1)

## plot heatmap along with dendrogram
plot(aggres1, sim)

## plot level with two clusters
plot(aggres1, x, k=2)

## run affinity propagation
apres <- apcluster(sim, q=0.7)

## create hierarchy of clusters determined by affinity propagation
aggres2 <- aggExCluster(sim, apres)
```

```
## show results
show(aggres2)

## plot dendrogram
plot(aggres2)

## plot heatmap
plot(aggres2, sim)

## plot level with two clusters
plot(aggres2, x, k=2)
```

AggExResult-class *Class "AggExResult"*

Description

S4 class for storing results of exemplar-based agglomerative clustering

Objects

Objects of this class can be created by calling [aggExCluster](#) for a given similarity matrix.

Slots

The following slots are defined for [AggExResult](#) objects:

l: number of samples in the data set

maxNoClusters: maximum number of clusters in the cluster hierarchy, i.e. it contains clusterings with 1 - maxNoClusters clusters.

exemplars: list of length maxNoClusters; the i-th component of the list is a vector of i exemplars (corresponding to the level with i clusters).

clusters: list of length maxNoClusters; the i-th component of clusters is a list of i clusters, each of which is a vector of sample indices.

merge: a maxNoClusters-1 by 2 matrix that contains the merging hierarchy; fully analogous to the slot merge in the class [hclust](#).

height: a vector of length maxNoClusters-1 that contains the merging objective of each merge; largely analogous to the slot height in the class [hclust](#) except that the slot height in AggExResult objects is supposed to be non-increasing, since [aggExCluster](#) is based on similarities, whereas [hclust](#) uses dissimilarities.

order: a vector containing a permutation of indices that can be used for plotting proper dendrograms without crossing branches; fully analogous to the slot order in the class [hclust](#).

labels: a character vector containing labels of clustered objects used for plotting dendrograms.

Methods

plot signature(x = "AggExResult"): see [plot-methods](#)

plot signature(x = "AggExResult", y = "matrix"): see [plot-methods](#)

show signature(object = "APResult"): see [show-methods](#)

cutree signature(object = "APResult", k="ANY", h="ANY"): see [cutree-methods](#)

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[aggExCluster](#), [show-methods](#), [plot-methods](#), [cutree-methods](#)

Examples

```
## create two Gaussian clouds
cl1 <- cbind(rnorm(50,0.2,0.05),rnorm(50,0.8,0.06))
cl2 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(cl1,cl2)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## compute agglomerative clustering from scratch
aggres1 <- aggExCluster(sim)

## show results
show(aggres1)

## plot dendrogram
plot(aggres1)

## plot heatmap along with dendrogram
plot(aggres1, sim)

## plot level with two clusters
plot(aggres1, x, k=2)

## run affinity propagation
apres <- apcluster(sim, q=0.7)

## create hierarchy of clusters determined by affinity propagation
aggres2 <- aggExCluster(sim, apres)
```

```
## show results
show(aggres2)

## plot dendrogram
plot(aggres2)

## plot heatmap
plot(aggres2, sim)

## plot level with two clusters
plot(aggres2, x, k=2)
```

apcluster

Affinity Propagation

Description

Runs affinity propagation clustering for a given similarity matrix

Usage

```
apcluster(s, p=NA, q=NA, maxits=1000, convits=100, lam=0.9,
          details=FALSE, nonoise=FALSE, seed=NA)
apclusterLM(s, p=NA, q=NA, maxits=1000, convits=100, lam=0.9,
            details=FALSE, nonoise=FALSE, seed=NA)
```

Arguments

s	an $l \times l$ similarity matrix
p	input preference; can be a vector that specifies individual preferences for each data point. If scalar, the same value is used for all data points. If NA, exemplar preferences are initialized according to the distribution of non-Inf values in s. How this is done is controlled by the parameter q.
q	if p=NA, exemplar preferences are initialized according to the distribution of non-Inf values in s. If q=NA, exemplar preferences are set to the median of non-Inf values in s. If q is a value between 0 and 1, the sample quantile with threshold q is used, whereas q=0.5 again results in the median.
maxits	maximal number of iterations that should be executed
convits	the algorithm terminates if the exemplars have not changed for convits iterations
lam	damping factor; should be a value in the range [0.5, 1); higher values correspond to heavy damping which may be needed if oscillations occur
details	if TRUE, more detailed information about the algorithm's progress is stored in the output object (see APResult)

nonoise	apcluster adds a small amount of noise to <code>s</code> to prevent degenerate cases; if TRUE, this is disabled
seed	for reproducibility, the seed of the random number generator can be set to a fixed value before adding noise (see above), if NA, the seed remains unchanged

Details

Affinity Propagation clusters data using a set of real-valued pairwise data point similarities as input. Each cluster is represented by a cluster center data point (the so-called exemplar). The method is iterative and searches for clusters maximizing an objective function called net similarity.

Apart from minor adaptations and optimizations, the implementation of the function `apclusterLM` is largely analogous to Frey's and Dueck's Matlab code (see <http://www.psi.toronto.edu/affinitypropagation/>). The function `apcluster` uses the same ideas, but replaces the loops in the computations of responsibilities and availabilities by pure matrix operations. For moderate data sets, the variant `apcluster` is approximately 60% faster than `apclusterLM`. For large data sets (several thousands of data samples), the use of `apclusterLM` (LM = Less Memory) may be advantageous, since this function requires less temporal storage (LM = Less Memory). For at most 5000 samples, we recommend to use `apcluster` (on up-to-date systems that are not too tight with memory).

The new argument `q` allows for better controlling the number of clusters without knowing the distribution of similarity values. A meaningful range for the parameter `p` can be determined using the function `preferenceRange`. Alternatively, a certain fixed number of clusters may be desirable. For this purpose, the function `apclusterK` is available.

Value

Upon successful completion, the function returns an `APResult` object.

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[APResult](#), [show-methods](#), [plot-methods](#), [labels-methods](#), [preferenceRange](#), [apclusterK](#)

Examples

```
## create two Gaussian clouds
cl1 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
cl2 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(cl1,cl2)

## create similarity matrix
sim <- negDistMat(x, r=2)

## run affinity propagation (p defaults to median of similarity)
apres <- apcluster(sim)

## show details of clustering results
show(apres)

## plot clustering result
plot(apres, x)

## plot heatmap
plot(apres, sim)

## run affinity propagation with default preference of 10% quantile
## of similarities; this should lead to a smaller number of clusters
apres <- apcluster(sim, q=0.1)
show(apres)
plot(apres, x)

## now try the same with RBF kernel
sim <- expSimMat(x, r=2)
apres <- apcluster(sim, q=0.2)
show(apres)
plot(apres, x)
```

apclusterDemo

Affinity Propagation Demo

Description

Runs affinity propagation demo for randomly generated data set according to Frey and Dueck

Usage

```
apclusterDemo(l=100, d=2, seed=NA, ...)
```

Arguments

l	number of data points to be generated
d	dimension of data to be created

seed	for reproducibility, the seed of the random number generator can be set to a fixed value; if NA, the seed remains unchanged
...	all other arguments are passed on to apcluster

Details

apclusterDemo creates 1 d-dimensional data points that are uniformly distributed in $[0, 1]^d$. Affinity propagation is executed for this data set with default parameters. Alternative settings can be passed to [apcluster](#) with additional arguments. After completion of affinity propagation, the results are shown and the performance measures are plotted.

This function corresponds to the demo function in the original Matlab code of Frey and Dueck. We warn the user, however, that uniformly distributed data are not necessarily ideal for demonstrating clustering, as there can never be real clusters in uniformly distributed data - all clusters found must be random artefacts.

Value

Upon successful completion, the function returns an invisible list with three components. The first is the data set that has been created, the second is the similarity matrix, and the third is an [APResult](#) object with the clustering results (see examples below).

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[APResult](#), [plot-methods](#), [apcluster](#)

Examples

```
## create random data set and run affinity propagation
apd <- apclusterDemo()

## plot clustering result along with data set
plot(apd[[3]], apd[[1]])
```

apclusterK

Affinity Propagation for Pre-defined Number of Clusters

Description

Runs affinity propagation clustering for a given similarity matrix adjusting input preferences iteratively in order to achieve a desired number of clusters

Usage

```
apclusterK(s, K, prc=10, bimaxit=20, exact=FALSE,
           nonoise=FALSE, seed=NA, verbose=FALSE, ...)
```

Arguments

s	an $l \times l$ similarity matrix
K	desired number of clusters; must be between 2 and $l-1$
prc	the algorithm stops if the number of clusters does not deviate more than prc percent from desired value K; set to 0 if you want to have exactly K clusters
bimaxit	maximum number of bisection steps to perform; note that no warning is issued if the number of clusters is still not in the desired range
exact	flag indicating whether or not to compute the initial preference range exactly (see preferenceRange)
nonoise	apcluster adds a small amount of noise to s to prevent degenerate cases; if TRUE, this is disabled
seed	for reproducibility, the seed of the random number generator can be set to a fixed value before adding noise (see above), if NA, the seed remains unchanged
verbose	flag indicating whether status information should be displayed during bisection
...	all other arguments are passed to apcluster as they are

Details

apclusterK first runs [preferenceRange](#) to determine the range of meaningful choices of the input preference p. Then it decreases p exponentially for a few iterations to obtain a good initial guess for p. If the number of clusters is still too far from the desired goal, bisection is applied.

Apart from minor adaptations and optimizations, the implementation is largely analogous to Frey's and Dueck's Matlab code (see <http://www.psi.toronto.edu/affinitypropagation/>).

Value

Upon successful completion, the function returns a [APResult](#) object.

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[apcluster](#), [preferenceRange](#), [APResult](#)

Examples

```
## create three Gaussian clouds
c11 <- cbind(rnorm(70, 0.2, 0.05), rnorm(70, 0.8, 0.06))
c12 <- cbind(rnorm(50, 0.7, 0.08), rnorm(50, 0.3, 0.05))
c13 <- cbind(rnorm(60, 0.8, 0.04), rnorm(60, 0.8, 0.05))
x <- rbind(c11, c12, c13)

## create similarity matrix
sim <- negDistMat(x, r=2)

## run affinity propagation such that 3 clusters are obtained
apres <- apclusterK(sim, 3)

## show details of clustering results
show(apres)

## plot clustering result
plot(apres, x)
```

APResult-class

Class "APResult"

Description

S4 class for storing results of affinity propagation clustering

Objects

Objects of this class can be created by calling [apcluster](#) for a given similarity matrix.

Slots

The following slots are defined for [APResult](#) objects. Most names are taken from Frey's and Dueck's original Matlab package:

l: number of samples in the data set

it: number of iterations the algorithm ran
p: input preference (either set by user or computed by [apcluster](#))
netsim: final total net similarity, defined as the sum of expref and dpsim (see below)
dpsim: final sum of similarities of data points to exemplars
expref: final sum of preferences of the identified exemplars
exemplars: vector containing indices of exemplars
clusters: list containing the clusters; the i-th component is a vector of indices of data points belonging to the i-th exemplar (including the exemplar itself)
idx: vector of length l realizing a sample-to-exemplar mapping; the i-th entry contains the index of the exemplar the i-th sample belongs to
netsimAll: vector containing the total net similarity for each iteration; only available if [apcluster](#) was called with details=TRUE
dpsimAll: vector containing the sum of similarities of data points to exemplars for each iteration; only available if [apcluster](#) was called with details=TRUE
exprefAll: vector containing the sum of preferences of the identified exemplars for each iteration; only available if [apcluster](#) was called with details=TRUE
idxAll: matrix with sample-to-exemplar indices for each iteration; only available if [apcluster](#) was called with details=TRUE

Methods

plot signature(x = "APResult"): see [plot-methods](#)
plot signature(x = "APResult", y = "matrix"): see [plot-methods](#)
show signature(object = "APResult"): see [show-methods](#)
labels signature(object = "APResult"): see [labels-methods](#)
cutree signature(object = "APResult"): see [cutree-methods](#)

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>
 Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).
 Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976.

See Also

[apcluster](#), [show-methods](#), [plot-methods](#), [labels-methods](#), [cutree-methods](#)

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## run affinity propagation
apres <- apcluster(sim, details=TRUE)

## show details of clustering results
show(apres)

## plot information about clustering run
plot(apres)

## plot clustering result
plot(apres, x)

## plot heatmap
plot(apres, sim)
```

ch22Promoters

ch22Promoters Data Set

Description

Toy data set consisting of promoter regions of 150 random genes from the human chromosome 22

Usage

```
ch22Promoters
```

Format

A character vector of length 150. Each string is a nucleotide sequence that corresponds to the promoter region of a gene from the human chromosome no. 22 (according to the human genome assembly hg18). The sequences start 999 bases upstream of the transcription start site (TSS) and end with the TSS itself. The names attribute contains the RefSeq IDs of the genes.

Examples

```
## load data set
data(ch22Promoters)

## display names of first five samples
names(ch22Promoters)[1:5]
```

```
## display last 50 bases of the first five samples
substr(ch22Promoters[1:5], 951, 1000)
```

cutree-methods

Cut Out Clustering Level from Cluster Hierarchy

Description

Cut out a clustering level from a cluster hierarchy

Usage

```
## S4 method for signature 'AggExResult'
cutree(tree, k, h)
## S4 method for signature 'APResult'
cutree(tree, k, h)
```

Arguments

tree	an object of class AggExResult containing a cluster hierarchy; may also be an object of class APResult
k	the level (i.e. the number of clusters) to be selected
h	alternatively, the level can be selected by specifying a cut-off for the merging objective

Details

The function `cutree` extracts a clustering level from a cluster hierarchy stored in an [AggExResult](#) object. Which level is selected can be determined by one of the two arguments `k` and `h` (see above). If both `k` and `h` are specified, `k` overrides `h`. This is done largely analogous to the standard function [cutree](#). The differences are (1) that only one level can be extracted at a time and (2) that an [ExClust](#) is returned instead of an index list.

The function `cutree` may further be used to convert an [APResult](#) object into an [ExClust](#) object. In this case, the arguments `k` and `h` are ignored.

Value

returns an object of class [ExClust](#)

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[AggExResult](#), [ExClust](#)

Examples

```
## create two simple clusters
x <- c(1, 2, 3, 7, 8, 9)
names(x) <- c("a", "b", "c", "d", "e", "f")

## compute similarity matrix (negative squared distance)
sim <- negDistMat(x, r=2)

## run affinity propagation
aggres <- aggExCluster(sim)

## show details of clustering results
show(aggres)

## retrieve clustering with 2 clusters
cutree(aggres, 2)

## retrieve clustering with cut-off h=-1
cutree(aggres, h=-1)
```

ExClust-class

Class "ExClust"

Description

S4 class for storing exemplar-based clusterings

Objects

Objects of this class can be created by calling [cutree](#) to cut out a clustering level from a cluster hierarchy of class [AggExResult](#). Moreover, [cutree](#) can also be used to convert an object of class [APResult](#) to class [ExClust](#).

Slots

The following slots are defined for [ExClust](#) objects:

l: number of samples in the data set

exemplars: vector containing indices of exemplars
clusters: list containing the clusters; the i-th component is a vector of indices of data points belonging to the i-th exemplar (including the exemplar itself)
idx: vector of length l realizing a sample-to-exemplar mapping; the i-th entry contains the index of the exemplar the i-th sample belongs to

Methods

plot signature(x = "ExClust", y = "matrix"): see [plot-methods](#)
show signature(object = "ExClust"): see [show-methods](#)
labels signature(object = "ExClust"): see [labels-methods](#)
cutree signature(object = "ExClust", k="ANY", h="ANY"): see [cutree-methods](#)

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>
 Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[aggExCluster](#), [show-methods](#), [plot-methods](#), [labels-methods](#), [cutree-methods](#), [AggExResult](#), [APResult](#)

Examples

```
## create two Gaussian clouds
cl1 <- cbind(rnorm(20,0.2,0.05),rnorm(20,0.8,0.06))
cl2 <- cbind(rnorm(25,0.7,0.08),rnorm(25,0.3,0.05))
x <- rbind(cl1,cl2)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## run affinity propagation
aggres <- aggExCluster(sim)

## extract level with two clusters
excl <- cutree(aggres, k=2)

## show details of clustering results
show(excl)

## plot information about clustering run
plot(excl, x)
```

labels-methods	<i>Generate label vector from clustering result</i>
----------------	---

Description

Generate a label vector from an clustering result

Usage

```
## S4 method for signature 'APResult'  
labels(object, type="names")  
## S4 method for signature 'ExClust'  
labels(object, type="names")
```

Arguments

object	object of class APResult or ExClust
type	specifies which kind of label vector should be created, see details below

Details

The function `labels` creates a label vector from a clustering result. Which kind of labels are produced is controlled by the argument `type`:

“**names**” (default) returns the name of the exemplar to which each data sample belongs to; if no names are available, the function stops with an error;

“**enum**” returns the index of the cluster to which each data sample belongs to, where clusters are enumerated consecutively from 1 to the number of clusters (analogous to other clustering methods like [kmeans](#));

“**exemplars**” returns the index of the exemplar to which each data sample belongs to, where indices of exemplars are within the original data, which is nothing else but the slot `object@idx` with attributes removed.

Value

returns a label vector as long as the number of samples in the original data set

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[APResult](#), [ExClust](#), [cutree](#)

Examples

```
## create two simple clusters
x <- c(1, 2, 3, 7, 8, 9)
names(x) <- c("a", "b", "c", "d", "e", "f")

## compute similarity matrix (negative squared distance)
sim <- negDistMat(x, r=2)

## run affinity propagation
apres <- apcluster(sim)

## show details of clustering results
show(apres)

## label vector (names of exemplars)
labels(apres)

## label vector (consecutive index of exemplars)
labels(apres, type="enum")

## label vector (index of exemplars within original data set)
labels(apres, type="exemplars")

## now with agglomerative clustering
aggres <- aggExCluster(sim)

## label (names of exemplars)
labels(cutree(aggres, 2))
```

plot-methods

Plot Clustering Results

Description

Functions for Visualizing Clustering Results

Usage

```
## S4 method for signature 'APResult,missing'
plot(x,
      type=c("netsim", "dpsim", "expref"),
      xlab="# Iterations",
      ylab="Similarity", ...)
## S4 method for signature 'APResult,matrix'
plot(x, y,
```

```

        connect=TRUE, xlab="", ylab="", ...)
## S4 method for signature 'ExClust,matrix'
plot(x, y,
      connect=TRUE, xlab="", ylab="", ...)
## S4 method for signature 'AggExResult,missing'
plot(x, main="Cluster dendrogram",
      xlab="", ylab="Balanced avg. similarity to exemplar",
      ticks=4, digits=2, ...)
## S4 method for signature 'AggExResult,matrix'
plot(x, y, k=NA, h=NA, ...)

```

Arguments

x	a clustering result object of class APResult , ExClust , or AggExResult
y	a matrix (see details below)
type	a string or array of strings indicating which performance measures should be plotted; valid values are "netsim", "dpsim", and "expref" which can be used in any combination or order; all other strings are ignored (for the meaning see APResult)
xlab	label for x axis of plot
ylab	label for y axis of plot
connect	used only if clustering is plotted on original data, ignored otherwise. If connect is TRUE, lines are drawn connecting exemplars with their cluster members.
main	title of plot
ticks	number of ticks used for the axis on the left side of the plot (applies to dendrogram plots only, see below)
digits	number of digits used for the axis tickmarks on the left side of the plot (applies to dendrogram plots only, see below)
k	level to be selected when plotting a single clustering level of cluster hierarchy (i.e. the number of clusters; see cutree-methods)
h	cut-off to be used when plotting a single clustering level of cluster hierarchy (see cutree-methods)
...	all other arguments are passed to the plotting command that are used internally, plot or heatmap .

Details

If `plot` is called for an [APResult](#) object without specifying the second argument `y`, a plot is created that displays graphs of performance measures over execution time of the affinity propagation run. This only works if `apcluster` was called with `details=TRUE`, otherwise an error message is displayed. This variant of `plot` does not return any value.

If `plot` is called for an [APResult](#) object along with a matrix as argument `y`, then the dimensions of the matrix determine the behavior of `plot`:

1. If the matrix `y` has two columns, `y` is interpreted as the original data set. Then a plot of the clustering result superimposed on the original data set is created. Each cluster is displayed in a different color. The exemplar of each cluster is highlighted by a black square. If `connect` is `TRUE`, lines connecting the cluster members to their exemplars are drawn. This variant of `plot` does not return any value.
2. If the matrix `y` is quadratic, `y` is interpreted as the similarity matrix that was used for clustering. Then a heatmap of the similarity matrix is drawn, where the clusters are grouped together and highlighted in different colors. The order of clusters is determined by running `aggExCluster` on the clustering result `x`. This variant of `plot` returns an invisible `AggExResult` object.

If `plot` is called for an `ExClust` object along with a matrix as argument `y`, then `plot` behaves exactly the same as described in the previous paragraph.

If `plot` is called for an `AggExResult` object without specifying the second argument `y`, then a dendrogram plot is drawn. This variant returns an invisible `dendrogram` object.

If `plot` is called for an `AggExResult` object along with a square matrix `y`, `y` is again interpreted as similarity matrix. If `x@maxNoClusters` is less than `x@1`, `plot` produces the same kind of heatmap as the variant for `APResult` and `ExClust` objects described above - with clusters grouped and highlighted in different colors. If `x@maxNoClusters` is equal to `x@1`, a heatmap is shown too, but instead of a color coding of clusters, the complete dendrogram of the cluster hierarchy is shown on the top and the left of the heatmap. This variant of `plot` does not return any value.

If `plot` is called for an `AggExResult` object along with a two-column matrix `y`, `y` is again interpreted as original data set. If one of the two arguments `k` or `h` is present, a clustering is cut out from the cluster hierarchy using `cutree` and this clustering is displayed with the original data set as described above. This variant of `plot` returns an invisible `ExClust` object containing the extracted clustering.

Value

see details above

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

See Also

[APResult](#), [AggExResult](#), [ExClust](#), [apcluster](#), [aggExCluster](#), [cutree-methods](#)

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(50,0.2,0.05),rnorm(50,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## run affinity propagation
apres <- apcluster(sim, q=0.6, details=TRUE)

## plot information about clustering run
plot(apres)

## plot clustering result
plot(apres, x)

## plot heatmap
plot(apres, sim)

## perform agglomerative clustering of affinity propagation clusters
aggres1 <- aggExCluster(sim, apres)

## show dendrogram
plot(aggres1)

## show heatmap
plot(aggres1, sim)

## show clustering result for 4 clusters
plot(aggres1, x, k=4)

## perform agglomerative clustering of whole data set
aggres2 <- aggExCluster(sim)

## show dendrogram
plot(aggres2)

## show heatmap along with dendrogram
plot(aggres2, sim)

## show clustering result for 2 clusters
plot(aggres2, x, k=2)
```

preferenceRange

Determine Meaningful Ranges for Input Preferences

Description

Determines meaningful ranges for affinity propagation input preference

Usage

```
preferenceRange(s, exact=FALSE)
```

Arguments

s	an $l \times l$ similarity matrix
exact	flag indicating whether exact ranges should be computed, which is relatively slow; if bounds are sufficient, supply FALSE (default)

Details

Affinity Propagation clustering relies on an appropriate choice of input preferences. This function helps in finding a good choice by determining meaningful lower and upper bounds.

Value

returns a vector with two entries, the first of which is the minimal input preference (which would lead to 1 or 2 clusters) and the second of which is the maximal input preference (which would lead to as many clusters as data samples)

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[apcluster](#)

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## create similarity matrix
sim <- negDistMat(x, r=2)

## determine bounds
preferenceRange(sim)
```

```
## determine exact range
preferenceRange(sim, exact=TRUE)
```

show-methods

Display Clustering Result Objects

Description

Display methods for S4 classes [APResult](#), [ExClust](#), and [AggExResult](#)

Usage

```
## S4 method for signature 'APResult'
show(object)
## S4 method for signature 'ExClust'
show(object)
## S4 method for signature 'AggExResult'
show(object)
```

Arguments

object an object of class [APResult](#), [ExClust](#), or [AggExResult](#)

Details

show displays the most important information stored in object.

For [APResult](#) objects, the number of data samples, the number of clusters, the number of iterations, the input preference, the final objective function values, the vector of exemplars, and the list of clusters are printed.

For [ExClust](#) objects, the number of data samples, the number of clusters, the vector of exemplars, and list of clusters are printed.

For [AggExResult](#) objects, only the number of data samples and the maximum number of clusters are printed. For retrieving a particular clustering level, use the function [cutree](#).

For accessing more detailed information, it is necessary to access the slots of object directly. Use [str](#) to get a compact overview of all slots of an object.

Value

show returns an invisible NULL

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

See Also

[APResult](#), [ExClust](#), [AggExResult](#), [cutree-methods](#)

Examples

```
## create two Gaussian clouds
c11 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
c12 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(c11,c12)

## compute similarity matrix (negative squared Euclidean)
sim <- negDistMat(x, r=2)

## run affinity propagation
apres <- apcluster(sim)

## show details of clustering results
show(apres)

## apply agglomerative clustering to apres
aggres <- aggExCluster(sim, apres)

## display overview of result
show(aggres)

## show clustering level with two clusters
show(cutree(aggres, 2))
```

similarity

Methods for Computing Similarity Matrices

Description

Compute similarity matrices from data set

Usage

```
negDistMat(x, r=1, ...)
expSimMat(x, r=2, w=1, ...)
linSimMat(x, w=1, ...)
linKernel(x, normalize=FALSE)
```

Arguments

x	real-valued data matrix; every row is a sample, every column a feature/input dimension
r	exponent (see details below)
w	radius (see details below)
normalize	see details below
...	all other arguments are passed to <code>dist</code> as they are; the default distance is <code>method="euclidean"</code> , see <code>dist</code> for other options.

Details

`negDistMat` creates a square matrix of mutual pairwise similarities of data vectors as negative distances. The argument `r` (default is 1) is used to transform the resulting distances by computing the `r`-th power (use `r=2` to obtain negative squared distances as in Frey's and Dueck's demos), i.e., given a distance `d`, the resulting similarity is computed as $s = -d^r$. Internally, the computation of distances is done using `dist`. All options of this function except `diag` and `upper` can be used, especially `method` which allows for selecting different distance measures.

`expSimMat` computes similarities in a way similar to `negDistMat`, but the transformation of distances to similarities is done in the following way:

$$s = \exp\left(-\left(\frac{d}{w}\right)^r\right)$$

As above, `r` is an exponent. The parameter `w` controls the speed of descent. `r=2` in conjunction with Euclidean distances corresponds to the well-known Gaussian/RBF kernel, whereas `r=1` corresponds to the Laplace kernel. Note that these similarity measures can also be understood as fuzzy equality relations.

`linSimMat` provides another way of transforming distances into similarities by applying the following transformation to a distance `d`:

$$s = \max\left(0, 1 - \frac{d}{w}\right)$$

Here `w` corresponds to a maximal radius of interest. Note that this is a fuzzy equality relation with respect to the Lukasiewicz t-norm.

Unlike the above three functions, `linKernel` computes pairwise similarities as scalar products of data vectors, i.e. it corresponds, as the name suggests, to the "linear kernel". If `normalize=TRUE`, the values are scaled to the unit sphere in the following way (for two samples `x` and `y`):

$$s = \frac{\vec{x}^T \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$$

Value

All functions listed above return square matrices of similarities.

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Micchelli, C. A. (1986) Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.* **2**, 11-20.

De Baets, B. and Mesiar, R. (1997) Pseudo-metrics and T-equivalences. *J. Fuzzy Math.* **5**, 471-481.

See Also

[dist](#), [apcluster](#)

Examples

```
## create two Gaussian clouds
cl1 <- cbind(rnorm(100,0.2,0.05),rnorm(100,0.8,0.06))
cl2 <- cbind(rnorm(50,0.7,0.08),rnorm(50,0.3,0.05))
x <- rbind(cl1,cl2)

## create negative distance matrix (default Euclidean)
sim1 <- negDistMat(x)

## compute similarities as squared negative distances
## (in accordance with Frey's and Dueck's demos)
sim2 <- negDistMat(x, r=2)

## compute RBF kernel
sim3 <- expSimMat(x, r=2)
```

sparseToFull

Convert a Sparse Similarity Matrix to Standard Representation

Description

Converts a similarity matrix in sparse format to a full l by l matrix

Usage

```
sparseToFull(s, fill=-Inf)
```

Arguments

s	a similarity matrix in sparse format; s must have exactly three columns; the matrix interpreted as follows: assume that a given row contains an integer index i in the 1st column and an integer index j in the 2nd column. Then the value in the 3rd column is interpreted as the similarity of data point i and data point j.
fill	value to fill in for entries not specified in s

Details

Frey's and Dueck's implementation can handle $l \times l$ similarity matrices and similarity matrices of the sparse format described above. Presently our implementation does not make any use of the sparsity of the similarity matrix. The functions `apcluster`, `apclusterK`, and `preferenceRange`, therefore, only accept square similarity matrices. This function is a utility for converting sparse similarity matrices - wherever they may come from.

Note that most of Frey's and Dueck's Matlab functions make an internal conversion. Guessing the right format from the matrix's dimension may be wrong for 3×3 matrices. We avoid this problem by explicitly distinguishing between sparse and dense similarity matrices (although we admit that clustering of 3 values makes as little sense as considering similarity matrices with only three entries).

Value

returns a square similarity matrix

Author(s)

Ulrich Bodenhofer & Andreas Kothmeier <apcluster@bioinf.jku.at>

References

<http://www.bioinf.jku.at/software/apcluster>

Frey, B. J. and Dueck, D. (2007) Clustering by passing messages between data points. *Science* **315**, 972-976. DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011) APCluster: an R package for affinity propagation clustering. *Bioinformatics* **27**, 2463-2464. DOI: [10.1093/bioinformatics/btr406](https://doi.org/10.1093/bioinformatics/btr406).

Examples

```
## create similarity matrix in sparse format
sp <- matrix(c(1, 1, 0.5, 3, 1, 0.2, 5, 4, -0.2, 3, 3, 1.2), 4, 3, byrow=TRUE)
sp

## perform conversion
s <- sparseToFull(sp, fill=0)
s
```

Index

*Topic **classes**

AggExResult-class, 6
APResult-class, 13
ExClust-class, 17

*Topic **cluster**

aggExCluster, 3
apcluster, 8
apclusterDemo, 10
apclusterK, 12
cutree-methods, 16
labels-methods, 19
plot-methods, 20
preferenceRange, 23
show-methods, 25
similarity, 26
sparseToFull, 28

*Topic **datasets**

ch22Promoters, 15

*Topic **methods**

cutree-methods, 16
labels-methods, 19
plot-methods, 20
show-methods, 25

*Topic **package**

apcluster-package, 2

aggExCluster, 2, 3, 6, 7, 18, 22
aggexcluster (aggExCluster), 3
AggExResult, 4–6, 16–18, 21, 22, 25, 26
AggExResult (AggExResult-class), 6
aggexresult (AggExResult-class), 6
AggExResult-class, 6
apcluster, 2, 8, 11–14, 21, 22, 24, 28, 29
apcluster-package, 2
apclusterDemo, 10
apclusterK, 9, 12, 29
apclusterLM (apcluster), 8
APResult, 2, 4, 8, 9, 11–13, 16–22, 25, 26
APResult (APResult-class), 13
apresult (APResult-class), 13

APResult-class, 13

ch22Promoters, 15

cutree, 16, 17, 20, 22, 25

cutree (cutree-methods), 16

cutree, AggExResult-method
(cutree-methods), 16

cutree, APResult-method
(cutree-methods), 16

cutree-methods, 5, 7, 14, 18, 21, 22, 26

cutree-methods, 16

dendrogram, 22

dist, 27, 28

ExClust, 4, 16, 17, 19–22, 25, 26

ExClust (ExClust-class), 17

exclust (ExClust-class), 17

ExClust-class, 17

expSimMat (similarity), 26

hclust, 4, 6

heatmap, 21

kmeans, 19

labels (labels-methods), 19

labels, APResult-method
(labels-methods), 19

labels, ExClust-method (labels-methods),
19

labels-methods, 9, 14, 18

labels-methods, 19

linKernel (similarity), 26

linSimMat (similarity), 26

negDistMat (similarity), 26

plot, 21

plot (plot-methods), 20

- plot, AggExResult, matrix-method
(plot-methods), [20](#)
- plot, AggExResult, missing-method
(plot-methods), [20](#)
- plot, APResult, matrix-method
(plot-methods), [20](#)
- plot, APResult, missing-method
(plot-methods), [20](#)
- plot, ExClust, matrix-method
(plot-methods), [20](#)
- plot-methods, [5](#), [7](#), [9](#), [11](#), [14](#), [18](#)
- plot-methods, [20](#)
- preferenceRange, [9](#), [12](#), [13](#), [23](#), [29](#)

- show (show-methods), [25](#)
- show, AggExResult-method (show-methods),
[25](#)
- show, APResult-method (show-methods), [25](#)
- show, ExClust-method (show-methods), [25](#)
- show-methods, [7](#), [9](#), [14](#), [18](#)
- show-methods, [25](#)
- similarity, [26](#)
- sparseToFull, [28](#)
- str, [25](#)