

Package ‘adjclust’

July 26, 2021

Maintainer Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>

Date 2021-07-26

Version 0.6.3

License GPL-3

Title Adjacency-Constrained Clustering of a Block-Diagonal Similarity Matrix

Description Implements a constrained version of hierarchical agglomerative clustering, in which each observation is associated to a position, and only adjacent clusters can be merged. Typical application fields in bioinformatics include Genome-Wide Association Studies or Hi-C data analysis, where the similarity between items is a decreasing function of their genomic distance. Taking advantage of this feature, the implemented algorithm is time and memory efficient. This algorithm is described in Ambroise et al (2019) <<https://almob.biomedcentral.com/articles/10.1186/s13015-019-0157-4>>.

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, Rcpp (>= 1.0.6), Matrix, sparseMatrixStats, methods, utils, capushe

Suggests knitr, testthat, rmarkdown, rioja, HiTC, snpStats, BiocGenerics

biocViews Clustering, FeatureExtraction

VignetteBuilder knitr

URL <https://github.com/pneuvial/adjclust>

BugReports <https://github.com/pneuvial/adjclust/issues>

RoxygenNote 7.1.1

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

Language en-US

NeedsCompilation yes

Author Christophe Ambroise [aut],
 Shubham Chaturvedi [aut],
 Alia Dehman [aut],
 Pierre Neuvial [aut, cre],
 Guillem Rigail [aut],
 Nathalie Vialaneix [aut],
 Gabriel Hoffman [aut]

Repository CRAN

Date/Publication 2021-07-26 19:30:07 UTC

R topics documented:

| | |
|--------------------|----|
| adjClust | 2 |
| chac | 4 |
| hicClust | 6 |
| plotSim | 8 |
| select | 9 |
| snpClust | 11 |

Index **13**

| | |
|----------|---|
| adjClust | <i>Adjacency-constrained Clustering</i> |
|----------|---|

Description

Adjacency-constrained hierarchical agglomerative clustering

Usage

```
adjClust(
  mat,
  type = c("similarity", "dissimilarity"),
  h = ncol(mat) - 1,
  strictCheck = TRUE
)
```

Arguments

| | |
|-------------|--|
| mat | A similarity matrix or a dist object. Most sparse formats from sparseMatrix are allowed |
| type | Type of matrix : similarity or dissimilarity. Defaults to "similarity" |
| h | band width. It is assumed that the similarity between two items is 0 when these items are at a distance of more than band width h. Default value is $\text{ncol}(\text{mat}) - 1$ |
| strictCheck | Logical (default to TRUE) to systematically check default of positivity in input similarities. Can be disabled to avoid computationally expensive checks when the number of features is large. |

Details

Adjacency-constrained hierarchical agglomerative clustering (HAC) is HAC in which each observation is associated to a position, and the clustering is constrained so as only adjacent clusters are merged. These methods are useful in various application fields, including ecology (Quaternary data) and bioinformatics (e.g., in Genome-Wide Association Studies (GWAS)).

This function is a fast implementation of the method that takes advantage of sparse similarity matrices (i.e., that have 0 entries outside of a diagonal band of width h). The method is fully described in (Dehman, 2015) and based on a kernel version of the algorithm. The different options for the implementation are available in the package vignette entitled "Notes on CHAC implementation in adjclust".

Value

An object of class `chac` which describes the tree produced by the clustering process. The object is a list with the same elements as an object of class `chac` (`merge`, `height`, `order`, `labels`, `call`, `method`, `dist.method`), and an extra element `mat`: the data on which the clustering is performed, possibly after pre-transformations described in the vignette entitled "Notes on CHAC implementation in adjclust".

References

Dehman A. (2015) *Spatial Clustering of Linkage Disequilibrium Blocks for Genome-Wide Association Studies*, PhD thesis, Universite Paris Saclay.

Ambroise C., Dehman A., Neuvial P., Rigail G., and Vialaneix N (2019). *Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics*, Algorithms for Molecular Biology 14(22)"

Randriamihamison, N., Vialaneix, N., & Neuvial, P. (2020). *Applicability and interpretability of Ward's hierarchical agglomerative clustering with or without contiguity constraints*, Journal of Classification 38, 1-27."

See Also

[snpClust](#) to cluster SNPs based on linkage disequilibrium

[hicClust](#) to cluster Hi-C data

Examples

```
sim <- matrix(
  c(1.0, 0.1, 0.2, 0.3,
    0.1, 1.0, 0.4, 0.5,
    0.2, 0.4, 1.0, 0.6,
    0.3, 0.5, 0.6, 1.0), nrow = 4)

## similarity, full width
fit1 <- adjClust(sim, "similarity")
plot(fit1)

## similarity, h < p-1
```

```

fit2 <- adjClust(sim, "similarity", h = 2)
plot(fit2)

## dissimilarity
dist <- as.dist(sqrt(2-(2*sim)))

## dissimilarity, full width
fit3 <- adjClust(dist, "dissimilarity")
plot(fit3)

## dissimilarity, h < p-1
fit4 <- adjClust(dist, "dissimilarity", h = 2)
plot(fit4)

```

chac

Class chac

Description

S3 class for Constrained Hierarchical Agglomerative Clustering results

Usage

```

## S3 method for class 'chac'
as.hclust(x, ...)

## S3 method for class 'chac'
print(x, ...)

## S3 method for class 'chac'
head(x, ...)

## S3 method for class 'chac'
summary(object, ...)

## S3 method for class 'chac'
plot(
  x,
  y,
  ...,
  mode = c("standard", "corrected", "total-disp", "within-disp", "average-disp"),
  nodeLabel = FALSE
)

diagnose(x, graph = TRUE, verbose = TRUE)

correct(x)

```

```
cutree_chac(tree, k = NULL, h = NULL)
```

Arguments

| | |
|-----------------|---|
| x, object, tree | an object of class 'chac' |
| ... | for <code>plot</code> , arguments passed to the function <code>plot.dendrogram</code> . Default values for <code>type</code> and <code>leaflab</code> are respectively set to "triangle" and "none" |
| y | not used |
| mode | type of dendrogram to plot (see Details). Default to "standard" |
| nodeLabel | (logical) whether the order of merging has to be displayed or not. <code>nodeLabel=TRUE</code> prints orders of fusion at corresponding nodes. Default to FALSE |
| graph | (logical) whether the diagnostic plot has to be displayed or not. Default to TRUE |
| verbose | (logical) whether to print a summary of the result or not. Default to TRUE |
| k | an integer scalar or vector with the desired number of groups |
| h | numeric scalar or vector with heights where the tree should be cut. Only available when the heights are increasing |

Details

Methods for class 'chac'

When `plot.chac` is called with `mode = "standard"`, the standard dendrogram is plotted, even though, due to contingency constrains, some branches are reversed (decreasing merges). When `plot.chac` is called with `mode = "corrected"`, a correction is applied to original heights so as to have only non decreasing merges). It does not change the result of the clustering, only the look of the dendrogram for easier interpretation.

Other modes are provided that correspond to different alternatives described in Grimm (1987):

- in `mode = "within-disp"`, heights correspond to within-cluster dispersion, *i.e.*, for a corresponding cluster, its height is

$$I(C) = \sum_{i \in C} d(i, g_C)$$

where d is the dissimilarity used to cluster objects and g_C is the center of gravity of cluster C . In this case, heights are always non decreasing;

- in `mode = "total-disp"`, heights correspond to the total within-cluster dispersion. It is obtained from `mode = "standard"` by the cumulative sum of its heights. In this case, heights are always non decreasing;
- in `mode = "average-disp"`, heights correspond to the within-cluster dispersion divided by the cluster size. In this case, there is no guaranty that the heights are non decreasing. When reversals are detected, a warning is printed to advice the user to change the mode of the representation.

Grimm (1987) indicates that heights as provided by `mode = "within-disp"` are highly dependent on cluster sizes and that the most advisable representation is the one provided by `mode = "total-disp"`. Further details are provided in the vignette "Notes on CHAC implementation in `adjclust`".

Value

The function `plot.chac` displays the dendrogram and additionally invisibly returns an object of class `dendrogram` with heights as specified by the user through the option `mode`.

`diagnose` invisibly exports a data frame with the numbers of decreasing merges described by the labels of the clusters being merged at this step and at the previous one, as well as the corresponding merge heights.

The function `correct` returns a `chac` objects with modified heights so as they are increasing. The new heights are calculated in an way identical to the option `mode = "corrected"` of the function `plot.chac` (see [Details](#)). In addition, the `chac` object has its field `method` modified from `adjClust` to `adjClust-modified`.

The function `cutree_chac` returns the clustering with `k` groups or with the groups obtained by cutting the tree at height `h`. If the heights are not increasing, the cutting of the tree is based on the corrected heights as provided by the function `correct`.

References

Grimm, E.C. (1987) CONISS: a fortran 77 program for stratigraphically constrained analysis by the method of incremental sum of squares. *Computer & Geosciences*, **13**(1), 13-35.

hicClust

Adjacency-constrained Clustering of Hi-C contact maps

Description

Adjacency-constrained hierarchical agglomerative clustering of Hi-C contact maps

Usage

```
hicClust(x, h = NULL, log = FALSE, ...)
```

Arguments

| | |
|------------------|--|
| <code>x</code> | either: 1. A <code>pxp</code> contact sparse or dense matrix (classes <code>matrix</code> , <code>Matrix</code> , <code>dscMatrix</code> , <code>dgTMatrix</code> , <code>dgCMatrix</code> , <code>dgeMatrix</code>). Its entries are the number of counts of physical interactions observed between all pairs of loci. 2. An object of class <code>HiTC::HTCexp</code> . The corresponding Hi-C data is stored as a <code>Matrix::dsCMatrix</code> object in the <code>intdata</code> slot. 3. A text file path with one line per pair of loci for which an interaction has been observed (in the format: <code>locus1<tab>locus2<tab>signal</code>) or a matrix or data frame with similar data (3 columns). |
| <code>h</code> | band width. If not provided, <code>h</code> is set to default value <code>'p-1'</code> . |
| <code>log</code> | logical. Whether to log-transform the count data. Default to <code>FALSE</code> . |
| <code>...</code> | further arguments to be passed to <code>read.table</code> function when <code>x</code> is a text file name. If not provided, the text file is supposed to be separated by tabulations, with no header. |

Details

Adjacency-constrained hierarchical agglomerative clustering (HAC) is HAC in which each observation is associated to a position, and the clustering is constrained so as only adjacent clusters are merged. Genomic regions (loci) are clustered according to information provided by high-throughput conformation capture data (Hi-C).

Value

An object of class `chac`.

References

Ambroise C., Dehman A., Neuvial P., Rigai G., and Vialaneix N (2019). *Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics*, Algorithms for Molecular Biology 14(22)"

Servant N. *et al* (2012). *HiTC : Exploration of High-Throughput 'C' experiments*. *Bioinformatics*.

See Also

[adjClust](#)

Examples

```
# input as HiTC::HTCexp object
## Not run:
if (require("HiTC", quietly = TRUE)) {
  load(system.file("extdata", "hic_imr90_40_XX.rda", package = "adjclust"))
  res1 <- hicClust(hic_imr90_40_XX)
}

## End(Not run)

# input as Matrix::dsCMatrix contact map
## Not run:
mat <- HiTC::intdata(hic_imr90_40_XX)
res2 <- hicClust(mat)

## End(Not run)

# input as text file
res3 <- hicClust(system.file("extdata", "sample.txt", package = "adjclust"))
```

plotSim

*Plot (dis)similarity matrix***Description**

Heatmap of the (dis)similarity matrix

Usage

```
plotSim(
  mat,
  type = c("similarity", "dissimilarity"),
  clustering = NULL,
  dendro = NULL,
  palette = heat.colors,
  breaks = 10,
  log = TRUE,
  h = NULL,
  stats = c("R.squared", "D.prime"),
  main = NULL,
  col.clust = "darkblue",
  lwd.clust = 2,
  xaxis = FALSE,
  naxis = 10
)
```

Arguments

| | |
|------------|--|
| mat | matrix to plot. It can be of class 'matrix', 'dgCMatrix', 'dsCMatrix', 'dist', 'HTCexp', 'snpMatrix'. |
| type | input matrix type. Can be either "similarity" or "dissimilarity" (kernels are supposed to be of type "similarity"). |
| clustering | vector of length the number of rows (columns) of the matrix that contains a contiguity constrained clustering (as provided by select for instance). If supplied the clustering is superimposed over the heatmap. |
| dendro | chac object as provided, e.g., by the function adjClust (or any of the other wrappers). |
| palette | color palette. Default to heat.colors |
| breaks | number of breaks used to set colors from the palette. Those are based on the quantiles of the matrix entries and for skewed distributions the actual number used to set the palette can be lower than breaks. |
| log | logical. Should the breaks be based on log-scaled values of the matrix entries. Default to TRUE. |
| h | if mat is of class "snpMatrix", band parameter used to compute the linkage disequilibrium (see ld). |

| | |
|-----------|---|
| stats | if mat is of class "snpMatrix", type of linkage disequilibrium measure (see 1d). |
| main | graphic title. |
| col.clust | color for the borders of the clusters (if clustering is provided). |
| lwd.clust | line width for the borders of the clusters (if clustering is provided). |
| xaxis | logical. Should a x-axis be displayed? Default to FALSE |
| naxis | number of breaks to display on the x-axis. For HTCexp objects, the axis is displayed in terms of Mpb and for the other types of input, it is displayed in terms of bin number. Default to 10. |

Details

This function produces a heatmap for the used (dis)similarity matrix that can be used as a diagnostic plot to check the consistency between the obtained clustering and the original (dis)similarity

See Also

[select](#), [adjClust](#)

Examples

```
# input as HiTC::HTCexp object
## Not run:
if (require("HiTC", quietly = TRUE)) {
  load(system.file("extdata", "hic_imr90_40_XX.rda", package = "adjclust"))
  plotSim(hic_imr90_40_XX)

  # with a constrained clustering
  res <- hicClust(hic_imr90_40_XX, log = TRUE)
  selected.capushe <- select(res)
  plotSim(hic_imr90_40_XX, clustering = selected.capushe, xaxis = TRUE)
  plotSim(hic_imr90_40_XX, clustering = selected.capushe, dendro = res)
}
## End(Not run)

plotSim(dist(iris[,1:4]), log = FALSE)
```

select

Clustering selection

Description

Clustering selection from a chac object with the slope heuristic or the broken stick heuristic

Usage

```
select(
  x,
  type = c("capushe", "bstick"),
  k.max = NULL,
  graph = FALSE,
  pct = 0.15
)
```

Arguments

| | |
|-------|---|
| x | an object of class 'chac' |
| type | model selection approach between slope heuristic ("capushe") and broken stick approach ("bstick") |
| k.max | maximum number of clusters that can be selected. Default to NULL, in which case it is set to $\min(\max(100, \frac{n}{\log(n)}), \frac{n}{2})$ where n is the number of objects to be clustered for capushe and to n for the broken stick model |
| graph | logical. Whether the diagnostic plot for the capushe selection is displayed or not. Default to FALSE |
| pct | minimum percentage of points for the plateau selection in capushe selection. See DDSE for further details |

Value

The function returns the clustering selected by the slope heuristic, as implemented in the R package capushe.

References

Baudry, J.P., Maugis, C. and Michel, B. (2012) Slope heuristics: overview and implementation. *Statistics and Computing*, **22**(2), 355-470. MacArthur, R.H. (1957) On the relative abundance of bird species. *Proceedings of the National Academy of Sciences*, **43**, 293-295.

Examples

```
## Not run: if (require("HiTC", quietly = TRUE)) {
  load(system.file("extdata", "hic_imr90_40_XX.rda", package = "adjclust"))
  res <- hicClust(hic_imr90_40_XX, log = TRUE)
  selected.capushe <- select(res)
  table(selected.capushe)
  selected.bs <- select(res, type = "bstick")
  table(selected.bs)
}
## End(Not run)

res <- adjClust(dist(iris[,1:4]))
select.clust <- select(res, "bs")
table(select.clust)
```

| | |
|----------|--|
| snpClust | <i>Adjacency-constrained Clustering of Single Nucleotide Polymorphisms</i> |
|----------|--|

Description

Adjacency-constrained hierarchical agglomerative clustering of Single Nucleotide Polymorphisms based on Linkage Disequilibrium

Usage

```
snpClust(x, h = ncol(x) - 1, stats = c("R.squared", "D.prime"))
```

Arguments

| | |
|-------|---|
| x | either a genotype matrix of class <code>SnpMatrix/matrix</code> or a linkage disequilibrium matrix of class <code>dgCMatrix</code> . In the latter case the LD values are expected to be in [0,1] |
| h | band width. If not provided, h is set to default value 'p-1' where 'p' is the number of columns of x |
| stats | a character vector specifying the linkage disequilibrium measures to be calculated (using the <code>ld</code> function) when x is a genotype matrix. Only "R.squared" and "D.prime" are allowed, see Details. |

Details

Adjacency-constrained hierarchical agglomerative clustering (HAC) is HAC in which each observation is associated to a position, and the clustering is constrained so as only adjacent clusters are merged. SNPs are clustered based on their similarity as measured by the linkage disequilibrium.

In the special case where genotypes are given as input and the corresponding LD matrix has missing entries, the clustering cannot be performed. This can typically happen when there is insufficient variability in the sample genotypes. In this special case, the indices of the SNP pairs which yield missing values are returned.

If x is of class `SnpMatrix` or `matrix`, it is assumed to be a $n \times p$ matrix of p genotypes for n individuals. This input is converted to a LD similarity matrix using the `snpStats::ld`. If x is of class `dgCMatrix`, it is assumed to be a (squared) LD matrix.

Clustering on a LD similarity other than "R.squared" or "D.prime" can be performed by providing the LD values directly as argument x. These values are expected to be in [0,1], otherwise they are truncated to [0,1].

Value

An object of class `chac` (when no LD value is missing)

References

Dehman A. (2015) *Spatial Clustering of Linkage Disequilibrium Blocks for Genome-Wide Association Studies*, PhD thesis, Universite Paris Saclay.

Dehman, A. Ambroise, C. and Neuvial, P. (2015). Performance of a blockwise approach in variable selection using linkage disequilibrium information. **BMC Bioinformatics** 16:148.

Ambroise C., Dehman A., Neuvial P., Rigail G., and Vialaneix N (2019). *Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics*, *Algorithms for Molecular Biology* 14(22)"

See Also

[adjClust ld](#)

Examples

```
## a very small example
if (requireNamespace("snpStats", quietly = TRUE)) {
  data(testdata, package = "snpStats")

  # input as snpStats::SnpMatrix
  fit1 <- snpClust(Autosomes[1:200, 1:5], h = 3, stats = "R.squared")

  # input as base::matrix
  fit2 <- snpClust(as.matrix(Autosomes[1:200, 1:5]), h = 3, stats = "R.squared")

  # input as Matrix::dgCMatrix
  ldres <- snpStats::ld(Autosomes[1:200, 1:5], depth = 3, stats = "R.squared", symmetric = TRUE)
  fit3 <- snpClust(ldres, 3)
}
```

Index

adjClust, [2](#), [7-9](#), [12](#)
as.hclust.chac (chac), [4](#)

chac, [3](#), [4](#), [7](#), [8](#), [11](#)
correct, [6](#)
correct (chac), [4](#)
cutree_chac, [6](#)
cutree_chac (chac), [4](#)
cuttree_chac (chac), [4](#)

DDSE, [10](#)
dendrogram, [6](#)
dgMatrix, [11](#)
diagnose, [6](#)
diagnose (chac), [4](#)

head.chac (chac), [4](#)
heat.colors, [8](#)
hicClust, [3](#), [6](#)

ld, [8](#), [9](#), [11](#), [12](#)

matrix, [11](#)

plot, [5](#)
plot.chac, [5](#)
plot.chac (chac), [4](#)
plot.dendrogram, [5](#)
plotSim, [8](#)
print.chac (chac), [4](#)

read.table, [6](#)

select, [8](#), [9](#), [9](#)
snpClust, [3](#), [11](#)
SnpMatrix, [11](#)
sparseMatrix, [2](#)
summary.chac (chac), [4](#)