

Package ‘clere’

March 22, 2016

Type Package

Title Simultaneous Variables Clustering and Regression

Version 1.1.4

Date 2016-03-18

Author Loic Yengo, Mickael Canouil

Maintainer Loic Yengo <loic.yengo@gmail.com>

Description Implements an empirical Bayes approach for simultaneous variable clustering and regression. This version also (re)implements in C++ an R script proposed by Howard Bondell that fits the Pairwise Absolute Clustering and Sparsity (PACS) methodology (see Sharma et al (2013) <DOI:10.1080/15533174.2012.707849>).

License GPL (>= 3)

Copyright Loic Yengo

Depends R (>= 2.10), parallel, methods, lasso2

Imports Rcpp

LinkingTo Rcpp, RcppEigen

Collate 'fitClere.R' 'Clere.R' 'sClere-Class.R' 'fitPacs.R'

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-03-22 20:20:14

R topics documented:

| | |
|----------------------------------|----|
| clere-package | 2 |
| algoComp-dataset | 3 |
| Clere-class | 4 |
| clusters | 6 |
| fitClere | 6 |
| fitPacs | 9 |
| numExpRealData-dataset | 10 |
| numExpSimData-dataset | 11 |
| Pacs-class | 12 |

| | |
|------------------------|----|
| plot-methods | 13 |
| predict | 14 |
| summary | 14 |

| | |
|--------------|-----------|
| Index | 16 |
|--------------|-----------|

| | |
|---------------|---|
| clere-package | <i>CLERE methodology for simultaneous variables clustering and regression</i> |
|---------------|---|

Description

The methodology consists in creating clusters of variables involved in a high dimensional linear regression model so as to reduce the dimensionality. A model-based approach is proposed and fitted using a Stochastic EM-Gibbs algorithm (SEM-Gibbs).

Details

```

Package:    clere
Title:      CLERE methodology for simultaneous variables clustering and regression
Version:    1.1.4
Date:       2016-03-18
Author:     Loic Yengo <loic.yengo@gmail.com>
Contributor: Mickael Canouil <mickael.canouil@cnrs.fr>
Maintainer: Loic Yengo <loic.yengo@gmail.com>
License:    GPL (>= 3)
Depends:    methods, parallel
Imports:    Rcpp
LinkingTo:  Rcpp, RcppEigen

```

The package implements mainly the `fitClere` function (an example is given below) for fitting the model from a matrix of covariates and a vector of response. The package also implements a `summary` method and graphical summary `plot` which represents the course of each parameters at each step of the SEM-Gibbs and a `predict` method for making prediction from a new design matrix.

Author(s)

Loic Yengo <loic.yengo@gmail.com>

References

Yengo L., Jacques J. and Biernacki C. Variable clustering in high dimensional linear regression, Journal de la Societe Francaise de Statistique (2013).

See Also

Overview : [clere-package](#)

Classes : [Clere](#)

Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)

Functions : [fitClere](#) Datasets : [numExpRealData](#), [numExpSimData](#)

Examples

```
# Simple example using simulated data
# to see how to you the main function clere
# library(clere)
x     <- matrix(rnorm(50 * 100), nrow = 50, ncol = 100)
y     <- rnorm(50)
model <- fitClere(y = y, x = x, g = 2, plotit = FALSE)
plot(model)
clus <- clusters(model, threshold = NULL)
predict(model, newx = x+1)
summary(model)
```

algoComp-dataset

Performances SEM algorithm versus MCEM

Description

This data contains four matrices corresponding to four performance indicators used to compare SEM algorithm and three versions of the MCEM algorithm (MCEMA: with 5 MC iterations; MCEMB: with 25 MC iterations and MCEMC: 125 MC iterations) as described in the package vignette. The first matrix Pred contains prediction errors; matrix Bias contains the bias over all model parameters, matrix Time contains execution times for the four methods and matrix Liks the log-likelihood reached by each method. These data were used to generate the Table 1. in the package vignette. For more details, please refer to the package vignette. The R script used to create this dataset is `clere/inst/doc/SEM_vs_MCEM_simulations.R`.

Usage

```
data(algoComp)
```

Format

A list containing four 200 x 4/5 matrices.

Author(s)

Loic Yengo <loic.yengol@gmail.com>

See Also

Overview : [clere-package](#)
 Classes : [Clere](#), [Pacs](#)
 Functions : [fitClere](#), [fitPacs](#)
 Datasets : [algoComp](#), [numExpSimData](#), [numExpRealData](#)

 Clere-class

[Clere class](#)

Description

This class contains all the input parameters to run CLERE.

Details

- y** [numeric]: The vector of observed responses.
- x** [matrix]: The matrix of predictors.
- n** [integer]: The sample size or the number of rows in matrix *x*.
- p** [integer]: The number of variables of the number of columns in matrix *x*.
- g** [integer]: The number or the maximum number of groups considered. Maximum number of groups stands when model selection is required.
- nItMC** [numeric]: Number of Gibbs iterations to generate the partitions.
- nItEM** [numeric]: Number of SEM/MCEM iterations.
- nBurn** [numeric]: Number of SEM iterations discarded before calculating the MLE which is averaged over SEM draws.
- dp** [numeric]: Number of iterations between sampled partitions when calculating the likelihood at the end of the run.
- nsamp** [numeric]: Number of sampled partitions for calculating the likelihood at the end of the run.
- sparse** [logical]: Should a \emptyset class be imposed to the model?
- analysis** [character]: Which analysis is to be performed. Values are "fit", "bic", "aic" and "icl".
- algorithm** [character]: The algorithm to be chosen to fit the model. Either the SEM-Gibbs algorithm or the MCEM algorithm. The most efficient algorithm being the SEM-Gibbs approach. MCEM is not available for binary response.
- initialized** [logical]: Is set to TRUE when an initial partition and an initial vector of parameters is given by the user.
- maxit** [numeric]: An EM algorithm is used inside the SEM to maximize the complete log-likelihood $p(y, Z | \theta)$. `maxit` stands as the maximum number of EM iterations for the internal EM.
- tol** [numeric]: Maximum increased in complete log-likelihood for the internal EM (stopping criterion).

- seed** [integer]: An integer given as a seed for random number generation. If set to NULL, then a random seed is generated between 1 and 1000.
- b** [numeric]: Vector of parameter b. Its size equals the number of group(s).
- pi** [numeric]: Vector of parameter pi. Its size equals the number of group(s).
- sigma2** [numeric]: Parameter σ^2 .
- gamma2** [numeric]: Parameter γ^2 . **itemintercept**[numeric]: Parameter β_0 (intercept).
- likelihood** [numeric]: Approximated log-likelihood.
- entropy** [numeric]: Approximated entropy.
- P** [matrix]: A $[p \times g]$ matrix of posterior probability of membership to the groups. $P = E[Z | \theta]$.
- theta** [matrix]: A $[nItEM \times (2g+4)]$ matrix containing values of the model parameters and complete data likelihood at each iteration of the SEM/MCEM algorithm
- Bw** [matrix]: A $[p \times nsamp]$ matrix which columns are samples from the posterior distribution of Beta (regression coefficients) given the data and the maximum likelihood estimates.
- Zw** [matrix]: A $[p \times nsamp]$ matrix which columns are samples from the posterior distribution of Z (groups membership indicators) given the data and the maximum likelihood estimates.
- theta0** [numeric]: A vector size $[2g+3]$ containing initial guess of the model parameters. See example for function [fitClere](#).
- Z0** [numeric]: A $[p \times 1]$ vector of integers taking values between 1 and p (number of variables).

Methods

- object["slotName"]**: Get the value of the field slotName.
- object["slotName" <-value]**: Set value to the field slotName.
- show(object)**: Returns the formatted values of [Clere](#) object.
- plot(x, ...)**: Graphical summary for MCEM/SEM-Gibbs estimation.
- clusters(object, threshold = NULL, ...)**: Returns the estimated clustering of variables.
- predict(object, newx, ...)**: Returns prediction using a fitted model and a new matrix of design.
- summary(object, ...)**: summarizes the output of function [fitClere](#).

Author(s)

Loic Yengo <loic.yengo@gmail.com>, Mickael Canouil <mickael.canouil@cnrs.fr>

See Also

Overview : [clere-package](#)
 Classes : [Clere](#)
 Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
 Functions : [fitClere](#), [fitPacs](#) Datasets : [numExpRealData](#), [numExpSimData](#)

| | |
|----------|------------------------|
| clusters | <i>clusters method</i> |
|----------|------------------------|

Description

This function makes returns the estimated clustering of variables.

Usage

```
## S4 method for signature 'Clere'
clusters(object, threshold = NULL, ...)
```

Arguments

| | |
|-----------|--|
| object | [Clere]: Output object from fitClere . |
| threshold | [numeric]: A numerical threshold > 0.5. If threshold = NULL then the each variable is assigned to the cluster having the largest associated posterior probability. |
| ... | Additional arguments, not to be supplied in this version. |

Author(s)

Loic Yengo <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)
 Classes : [Clere](#)
 Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
 Functions : [fitClere](#)

| | |
|----------|--------------------------|
| fitClere | <i>fitClere function</i> |
|----------|--------------------------|

Description

This function runs the CLERE Model. It returns an object of class [Clere](#). For more details please refer to [clere](#).

Usage

```
fitClere(y = rnorm(10), x = matrix(rnorm(50), nrow = 10),
         g = 1, nItMC = 50, nItEM = 1000, nBurn = 200,
         dp = 5, nsamp = 200, maxit = 500, tol = 0.001,
         nstart=2, parallel = FALSE, seed = NULL, plotit = FALSE, sparse = FALSE,
         analysis = "fit", algorithm = "SEM", theta0 = NULL, Z0 = NULL)
```

Arguments

| | |
|------------------------|---|
| <code>y</code> | [numeric]: The vector of observed responses - size n . |
| <code>x</code> | [matrix]: The matrix of predictors - size n rows and p columns. |
| <code>g</code> | [integer]: Either the number or the maximum of groups for fitting CLERE. Maximum number of groups is considered when model selection is required. |
| <code>nItMC</code> | [numeric]: Number of Gibbs iterations to generate the partitions. After the <code>nBurn</code> iterations, this number is automatically set to 1. |
| <code>nItEM</code> | [numeric]: Number of SEM iterations. |
| <code>nBurn</code> | [numeric]: Number of SEM iterations discarded before calculating the MLE which is averaged over SEM draws. |
| <code>dp</code> | [numeric]: Number of iterations between sampled partitions when calculating the likelihood at the end of the run. |
| <code>nsamp</code> | [numeric]: Number of sampled partitions for calculating the likelihood at the end of the run. |
| <code>maxit</code> | [numeric]: An EM algorithm is used inside the SEM to maximize the complete log-likelihood $p(y, Z \theta)$. <code>maxit</code> stands as the maximum number of EM iterations for the internal EM. |
| <code>tol</code> | [numeric]: Maximum increased in complete log-likelihood for the internal EM (stopping criterion). |
| <code>nstart</code> | [integer]: Number of random starting points to be used for fitting the model. |
| <code>parallel</code> | [logical]: Should the estimation from <code>nstart</code> random starting points run in parallel? |
| <code>seed</code> | [integer]: An integer given as a seed for random number generation. If set to NULL, then a random seed is generated between 1 and 1000. |
| <code>plotit</code> | [logical]: Should a summary plot (base plot) be drawn after the run? |
| <code>sparse</code> | [logical]: Should a \emptyset class be imposed to the model? |
| <code>analysis</code> | [character]: Which analysis is to be performed. Values are "fit", "bic", "aic" and "icl". |
| <code>algorithm</code> | [character]: The algorithm to be chosen to fit the model. Either the SEM-Gibbs algorithm or the MCEM algorithm. The most efficient algorithm being the SEM-Gibbs approach. MCEM is not available for binary response. |
| <code>theta0</code> | [vector(numeric)]: An initial guess of the model parameters. When considering g components, the length of <code>theta0</code> must be $2*g+3$ and <code>theta0</code> should be filled as intercept, the b_k 's (g real numbers), the π_k 's (g real numbers summing to 1), σ^2 and γ^2 (two positive numbers). |
| <code>Z0</code> | [vector(integer)]: A vector of integers representing an initial partition for the variables. For 10 variables and 3 groups <code>Z0</code> can be defined as <code>c(rep(0, 2), rep(1, 3), rep(2, 5))</code> . |

Value

Object of class `Clere`.

Author(s)

Loic Yengo <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)
 Classes : [Clere](#)
 Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
 Functions : [fitClere](#) [fitPacs](#)

Examples

```
# library(clere)
plotit <- FALSE
sparse <- FALSE
nItEM <- 100
nBurn <- nItEM / 2
nsamp <- 100
analysis <- "fit"
algorithm <- "SEM"
nItMC <- 1
dp <- 2
maxit <- 200
tol <- 1e-3

n <- 50
p <- 50
intercept <- 0
sigma <- 10
gamma <- 10
rho <- 0.5

g <- 5
probs <- c(0.36, 0.28, 0.20, 0.12, 0.04)
Eff <- p * probs
a <- 5
B <- a*(0:(g-1))-1
Z <- matrix(0, nrow = p, ncol = g)
imax <- 0
imin <- 1

for (k in 1:g) {
  imin <- imax+1
  imax <- imax+Eff[k]
  Z[imin:imax, k] <- 1
}
Z <- Z[sample(1:p, p), ]
if (g>1) {
  Beta <- rnorm(p, mean = c(Z%*%B), sd = gamma)
} else {
  Beta <- rnorm(p, mean = B, sd = gamma)
}
```



```

theta0 <- NULL # c(intercept, B, probs, sigma^2, gamma^2)
Z0      <- NULL # apply(Z, 1, which.max)-1

gmax <- 7

## Prediction
eps  <- rnorm(n, mean = 0, sd = sigma)
X    <- matrix(rnorm(n*p), nrow = n, ncol = p)
Y    <- as.numeric(intercept+X%%Beta+eps)
tt   <- system.time(mod <- fitClere(y = Y, x = X, g = gmax,
                                   analysis = analysis, algorithm = algorithm,
                                   plotit = plotit,
                                   sparse = FALSE, nItEM = nItEM,
                                   nBurn = nBurn, nItMC = nItMC,
                                   nsamp = nsamp, theta0 = theta0, Z0 = Z0) )

plot(mod)
Yv <- predict(object = mod, newx = X)

```

fitPacs

fitPacs function

Description

This function implements the PACS (Pairwise Absolute Clustering and Sparsity) methodology of Sharma DB et al. (2013). This methodology proposes to estimate the regression coefficients by solving a penalized least squares problem. It imposes a constraint on Beta (the vector of regression coefficients) that is a weighted combination of the L1 norm and the pairwise L-infinity norm. Upper-bounding the pairwise L-infinity norm enforces the covariates to have close coefficients. When the constraint is strong enough, closeness translates into equality achieving thus a grouping property. For PACS, no software was available. Only an R script was released on Bondell's webpage (<http://www4.stat.ncsu.edu/~bondell/Software/PACS/PACS.R.r>). Since this R script was running very slowly, we decided to reimplement it in C++ and interfaced it with the present R package clere. This corresponds to the option type=1 in Bondell's script.

Usage

```

fitPacs(Y = rnorm(10), X = matrix(rnorm(50), nrow = 10),
        lambda=0.5, betaInput=rnorm(10), epsPACS=1e-5, nItMax=1000)

```

Arguments

| | |
|--------|--|
| Y | [numeric]: The vector of observed responses - size n. |
| X | [matrix]: The matrix of predictors - size n rows and p columns. |
| lambda | [numeric]: A non-negative penalty term that controls simultaneously clustering and sparsity. |

| | |
|-----------|--|
| betaInput | [numeric]: A vector of initial guess of the model parameters. The authors suggest to use coefficients obtained after fitting a ridge regression with the shrinkage parameter selected using AIC criterion. |
| epsPACS | [numeric]: A tolerance threshold that control the convergence of the algorithm. The default value fixed in Bondell's initial script is 1e-5. |
| nItMax | [numeric]: Maximum number of iterations in the algorithm. |

Value

Object of class `Pacs` containing all the input parameters plus parameter a_0 the intercept and parameter K the dimensionality of the model.

Author(s)

Loic Yengo (re-implementation of Bondell's script) <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)
 Classes : [Clere](#)
 Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
 Functions : [fitClere](#) [fitPacs](#)

Examples

```
n <- 100
p <- 20
Beta <- rep(c(0,2),10)
eps <- rnorm(n,sd=3)
x <- matrix(rnorm(n*p), nrow = n, ncol = p)
y <- as.numeric(10+x%*%Beta+eps)
bInit <- lm(y~scale(x))$coefficients[-1]
mod <- fitPacs(Y=y,X=x,lambda=1.25,betaInput=bInit,epsPACS=1e-5,nItMax=1000)
```

numExpRealData-dataset

Performances of 9 methods for dimension reduction applied to 2 published real dataset

Description

This data contains two matrices: one for the Prostate dataset (from R package `lasso2`) and the other for the `eyedata` dataset (from R package `flare`). Each matrix has 5 rows and 28 columns. The columns can be grouped as three blocs of 9 (for each method compared: LASSO, RIDGE, Elastic net [ELNET], Stepwise variable selection [STEP], CLERE, CLERE sparse [CLERE_s], Spike and Slab [SS], AVG method and Pairwise Absolute Clustering and Sparsity [PACS]). The 1st 9 (1:9) contain prediction error obtained by 5-fold cross validation using 10 random permutation of the covariate

matrix. The 2nd 9 columns (10:18) contain the number of parameters estimated for each method. The 3rd 9 columns are times in seconds measured for fitting each methods. The 28 column is the seed utilized for generating random numbers in these analyses. For more details, please refer the package vignette. The R script used to create this dataset is `clere/inst/doc/RealDataExample.R`.

Usage

```
data(numExpRealData)
```

Format

A list containing two matrices: one for the Prostate dataset (from R package `lasso2`) and the other for the Eye dataset (from R package `flare`)

Author(s)

Loic Yengo <loic.yengol@gmail.com>

See Also

Overview : [clere-package](#)

Classes : [Clere](#), [Pacs](#)

Functions : [fitClere](#), [fitPacs](#)

Datasets : [numExpSimData](#), [numExpRealData](#), [algoComp](#)

numExpSimData-dataset *Performances of 9 methods for dimension reduction on data simulated under the CLERE model*

Description

This dataset is a matrix of 200 rows and 28 columns. The columns can be grouped as three blocs of 9 (for each method compared: LASSO, RIDGE, Elastic net [ELNET], Stepwise variable selection [STEP], CLERE, CLERE sparse [CLERE_s], Spike and Slab [SS], AVG method and Pairwise Absolute Clustering and Sparsity [PACS]). Prediction errors (MSE), number of estimated parameters and time (seconds) to fit the data are compared. The 1st 9 (1:9) contain prediction error obtained by 5-fold cross validation using 10 random permutation of the covariate matrix. The 2nd 9 columns (10:18) contain the number of parameters estimated for each method. The 3rd 9 columns are times in seconds measured for fitting each methods. The 28 column is the seed utilized for generating random numbers in these analyses. Each row corresponds to a simulated dataset on which all 9 methods were fitted. For more details, please refer to the package vignette. The R script used to create this dataset is `clere/inst/doc/SimulatedDataExample.R`.

Usage

```
data(numExpSimData)
```

Format

A 200 x 28 matrix.

Author(s)

Loic Yengo <loic.yengol@gmail.com>

See Also

Overview : [clere-package](#)

Classes : [Clere](#), [Pacs](#)

Functions : [fitClere](#), [fitPacs](#)

Datasets : [numExpRealData](#), [numExpSimData](#), [algoComp](#)

Pacs-class

[Pacs class](#)

Description

This class contains all the input parameters to run CLERE.

Details

Y [numeric]: The vector of observed responses - size n.

X [matrix]: The matrix of predictors - size n rows and p columns.

lambda [numeric]: A non-negative penalty term that controls simultaneously clustering and sparsity.

betaInput [numeric]: A vector of initial guess of the model parameters. The authors suggest to use coefficients obtained after fitting a ridge regression with the shrinkage parameter selected using AIC criterion.

epsPACS [numeric]: A tolerance threshold that control the convergence of the algorithm. The default value fixed in Bondell's initial script is 1e-5.

nItMax [integer]: Maximum number of iterations in the algorithm.

a0 [numeric]: Fitted intercept.

K [integer]: Model dimensionality.

Methods

object[*"slotName"*] : Get the value of the field *slotName*.

object[*"slotName"*] <-value: Set value to the field *slotName*.

Author(s)

Loic Yengo <loic.yengo@gmail.com>, Mickael Canouil <mickael.canouil@cnrs.fr>

See Also

Overview : [clere-package](#)
Classes : [Clere](#), [Pacs](#)
Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
Functions : [fitClere](#), [fitPacs](#)

plot-methods

plot method

Description

Graphical summary for MCEM/SEM-Gibbs estimation. This function represents the course of the model parameters in view of the iterations of the estimation algorithms implemented in [fitClere](#).

Usage

```
## S4 method for signature 'Clere'  
plot(x, y, ...)
```

Arguments

x [Clere]: Output object from [fitClere](#).
y [any]: Unused parameter.
... Additional arguments, not to be supplied in this version.

Author(s)

Loic Yengo <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)
Classes : [Clere](#)
Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
Functions : [fitClere](#)

predict *predict method*

Description

This function makes prediction using a fitted model and a new matrix of design. It returns a vector of predicted values of size equal to the number of rows of matrix `newx`.

Usage

```
## S4 method for signature 'Clere'  
predict(object, newx, ...)
```

Arguments

| | |
|---------------------|---|
| <code>object</code> | [Clere]: Output object from fitClere . |
| <code>newx</code> | [matrix]: A numeric design matrix. |
| <code>...</code> | Additional arguments, not to be supplied in this version. |

Author(s)

Loic Yengo <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)
Classes : [Clere](#)
Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)
Functions : [fitClere](#)

summary *summary method*

Description

This function summarizes the output of function [fitClere](#).

Usage

```
## S4 method for signature 'Clere'  
summary(object, ...)
```

Arguments

| | |
|---------------------|---|
| <code>object</code> | [Clere]: Output object from fitClere . |
| <code>...</code> | Additional arguments, not to be supplied in this version. |

Author(s)

Loic Yengo <loic.yengo@gmail.com>

See Also

Overview : [clere-package](#)

Classes : [Clere](#)

Methods : [show](#), [plot](#), [clusters](#), [predict](#), [summary](#)

Functions : [fitClere](#)

Index

*Topic **Clere**

- Clere-class, 4
- clere-package, 2
- clusters, 6
- fitClere, 6
- fitPacs, 9
- Pacs-class, 12
- plot-methods, 13
- predict, 14
- summary, 14

*Topic **Pacs**

- Pacs-class, 12

*Topic **algoComp**

- algoComp-dataset, 3

*Topic **class**

- Clere-class, 4
- Pacs-class, 12

*Topic **clere-package**

- clere-package, 2

*Topic **clere**

- algoComp-dataset, 3
- clere-package, 2
- clusters, 6
- fitClere, 6
- fitPacs, 9
- numExpRealData-dataset, 10
- numExpSimData-dataset, 11
- plot-methods, 13
- predict, 14
- summary, 14

*Topic **clusters**

- clusters, 6

*Topic **datasets**

- algoComp-dataset, 3
- numExpRealData-dataset, 10
- numExpSimData-dataset, 11

*Topic **fitClere**

- fitClere, 6
- fitPacs, 9

*Topic **fitPacs**

- fitPacs, 9

*Topic **function**

- fitClere, 6
- fitPacs, 9

*Topic **methods**

- Clere-class, 4
- clusters, 6
- Pacs-class, 12
- plot-methods, 13
- predict, 14
- summary, 14

*Topic **method**

- Clere-class, 4
- clusters, 6
- Pacs-class, 12
- plot-methods, 13
- predict, 14
- summary, 14

*Topic **numExpRealData**

- numExpRealData-dataset, 10

*Topic **numExpSimData**

- numExpSimData-dataset, 11

*Topic **package**

- clere-package, 2

*Topic **plot**

- plot-methods, 13

*Topic **summary**

- predict, 14
- summary, 14

[, Clere-method (Clere-class), 4

[, sClere-method (Clere-class), 4

[<- , Clere-method (Clere-class), 4

[<- , sClere-method (Clere-class), 4

algoComp, 4, 11, 12

algoComp (algoComp-dataset), 3

algoComp-dataset, 3

Clere, 3–8, 10–15

clere, 6
clere (clere-package), 2
Clere-class, 4
clere-package, 2
clusters, 3, 5, 6, 6, 8, 10, 13–15
clusters, Clere-method (clusters), 6
clusters-methods (clusters), 6

fitClere, 2–6, 6, 8, 10–15
fitPacs, 4, 5, 8, 9, 10–13

numExpRealData, 3–5, 11, 12
numExpRealData
 (numExpRealData-dataset), 10
numExpRealData-dataset, 10
numExpSimData, 3–5, 11, 12
numExpSimData (numExpSimData-dataset),
 11
numExpSimData-dataset, 11

Pacs, 4, 10–13
Pacs-class, 12
plot, 2, 3, 5, 6, 8, 10, 13–15
plot (plot-methods), 13
plot, Clere, ANY-method (plot-methods), 13
plot, Clere-method (plot-methods), 13
plot-methods, 13
predict, 2, 3, 5, 6, 8, 10, 13, 14, 14, 15
predict, Clere-method (predict), 14
predict-methods (predict), 14

sClere-class (Clere-class), 4
show, 3, 5, 6, 8, 10, 13–15
show, Clere-method (Clere-class), 4
show, sClere-method (Clere-class), 4
summary, 2, 3, 5, 6, 8, 10, 13, 14, 14, 15
summary, Clere-method (summary), 14
summary-methods (summary), 14