

Package ‘Qbone’

May 12, 2024

Title Tools for Quantile Functional Regression with Quantlets

Version 0.0.0.9100

URL <https://github.com/siminpan/Qbone>

BugReports <https://github.com/siminpan/Qbone/issues>

Description The ‘Qbone’ package is a set of tools that enables efficient estimation of quantlets used in quantile functional regression to analyze distributional data. This package can be used to analyze distributional data in broader applications such as climate change analysis, image recognition analysis, and personalized health data analysis, etc. At current stage, it is just for independent functional linear regression.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

LinkingTo Rcpp,
RcppEigen

Imports Rcpp,
methods,
data.table,
stats,
utils,
pbapply,
glmnet,
doMC,
Matrix,
foreach,
MASS,
ggplot2,
gridExtra,
pracma,
wavethresh,
plotly,
magrittr,

dplyr,
grDevices,
RColorBrewer

Collate 'Qbone-package.R'

'utils.R'
'zzz.R'
'generics.R'
'QboneData.R'
'QboneObject.R'
'RcppExports.R'
'basisfunction.R'
'functionalregression.R'
'plotting.R'
'preprocessing.R'

R topics documented:

Qbone-package	3
addMetaData	3
assayData	4
assays	5
createQboneData	5
createQboneObject	6
defaultAssay	8
dxPlot	9
dxPlotRev	9
ecQuantlets	10
eigenmm	10
histogram3D	11
idents	12
lassoList	13
pdPlot	14
preQuantlets	15
qbasisPlot	16
qbasisPlot3D	16
Qbone-class	17
Qbone-methods	17
QboneData-class	18
QboneData-methods	18
qfrModel	19
quantileFPlot3D	20
readQbone	21
samples	21
set-if-null	22
thinData	22

Index

Qbone-package

Qbone: Tools for Quantile Functional Regression with Quantlets

Description

The ‘Qbone’ package is a set of tools that enables efficient estimation of quantlets used in quantile functional regression to analyze distributional data. This package can be used to analyze distributional data in broader applications such as climate change analysis, image recognition analysis, and personalized health data analysis, etc. At current stage, it is just for independent functional linear regression.

Author(s)

Maintainer: Simin Pan <simonpan@tamu.edu> ([ORCID](#))

Authors:

- Linying Zhang <zly.rebecca@tamu.edu> ([ORCID](#))

Other contributors:

- Alan Dabney <adabney@stat.tamu> ([ORCID](#)) [thesis advisor]
- Carl Gregory <cgregory@tamu.edu> ([ORCID](#)) [thesis advisor]

See Also

Useful links:

- <https://github.com/siminpan/Qbone>
- Report bugs at <https://github.com/siminpan/Qbone/issues>

addMetaData

Add in metadata associated

Description

Add in metadata associated

Usage

```
addMetaData(object, metadata, col.name = NULL)
```

```
## S3 method for class 'Qbone'
```

```
addMetaData(object, metadata, col.name = NULL)
```

Arguments

object	An object
metadata	A vector, list, or data.frame with metadata to add
col.name	A name for meta data if not a named list or data.frame

Value

object with metadata added

Examples

```
## Not run:
meta.data2 = data.frame(group2 = c("a2", "a2", "b2", "b2"))
qbone1 <- addMetaData(object = qbone1, metadata = meta.data2[,1], col.name = "group2")
# or
qbone1[["group2"]] <- meta.data2[,1]

## End(Not run)
```

assayData

Get and Set Assay Data

Description

General accessor and setter functions for [QboneData](#) objects. `getQboneData` can be used to pull information from any of the expression matrices (eg. “counts”, “data”, or “scale.data”). `SetAssayData` can be used to replace one of these expression matrices || double check

Usage

```
getQboneData(object, slot, ...)
```

```
## S3 method for class 'Qbone'
getQboneData(object, slot = "data", assay = NULL, ...)
```

```
## S3 method for class 'QboneData'
getQboneData(object, slot = c("data", "scale.data"), ...)
```

Arguments

object	An object
slot	Specific assay data to get or set
...	Arguments passed to other methods
assay	Specific assay to get data from or set data for; defaults to the default assay

Value

getQboneData: returns the specified assay data

Examples

```
## Not run:
# Get the data directly from an QboneData object
getQboneData(qbone1[["Bone"]], slot = "data")

## End(Not run)
```

assays	<i>Query Specific Object Types</i>
--------	------------------------------------

Description

Adopted from Seurateobject package List the names of [assay](#)

Usage

```
assays(object, slot = NULL)
```

Arguments

- object A [Qbone](#) object
- slot Name of component object to return

Value

If slot is NULL, the names of all component objects in this Qbone object. Otherwise, the specific object specified

createQboneData	<i>create an QboneData object</i>
-----------------	-----------------------------------

Description

create an QboneData object

Usage

```
createQboneData(
  data,
  meta.assays = NULL,
  sampleid.assays = 1,
  assay.name = "Bone",
  assay.orig = NULL,
  sort = F,
  ...
)
```

Arguments

<code>data</code>	Sample data
<code>meta.assays</code>	metadata for the assay with sample names
<code>sampleid.assays</code>	column number of sample name in meta.assay
<code>assay.name</code>	assay name for the data.
<code>assay.orig</code>	Original assay that this assay is based off of. Used to track assay provenance
<code>sort</code>	sort the data before put into data slot if T, default is F.

Value

A `QboneData` object

Examples

```
## Not run:
n = 10000
list1 = list(rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0.5, sd = 1),
             rnorm(n, mean = 0.5, sd = 1))
meta.data = data.frame(name = c("a_1", "a_2", "b_1", "b_2"), group = c("a", "a", "b", "b"))
rownames(meta.data) = meta.data[,1]
qbonedata = createQboneData(list1, meta.data, sampleid = 1)

## End(Not run)
```

<code>createQboneObject</code>	<i>Create a Qbone object</i>
--------------------------------	------------------------------

Description

Create a `Qbone` object

Usage

```

createQboneObject(
  data,
  project = "createQboneObject",
  assay = "Bone",
  names.field = 1,
  names.delim = "_",
  meta.data = NULL,
  sampleid = 1,
  ...
)

## Default S3 method:
createQboneObject(
  data = data,
  project = "QboneProject",
  assay = "Bone",
  names.field = 1,
  names.delim = "_",
  meta.data = NULL,
  sampleid = 1,
  ...
)

## S3 method for class 'QboneData'
createQboneObject(
  data = data,
  project = "QboneProject",
  assay = "Bone",
  names.field = 1,
  names.delim = "_",
  meta.data = NULL,
  sampleid = 1,
  ...
)

```

Arguments

data	data
project	project name for the Qbone object
assay	Name of the initial assay
names.field	number of field of names
names.delim	delimiter of names
meta.data	Include cells where at least this many features are detected.
sampleid	column number of sample name in mate.data

Examples

```
## Not run:
n = 10000
list1 = list(rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0.5, sd = 1),
             rnorm(n, mean = 0.5, sd = 1))
meta.data = data.frame(name = c("a_1", "a_2", "b_1", "b_2"),
                       group = c("a", "a", "b", "b"))
rownames(meta.data) = meta.data[,1]
qbonedata = createQboneData(list1, meta.data, sampleid = 1)
qbone1 = createQboneObject(qbonedata, meta.data = meta.data)

## End(Not run)

## Not run:
n = 10000
list1 = list(rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0, sd = 1),
             rnorm(n, mean = 0.5, sd = 1),
             rnorm(n, mean = 0.5, sd = 1))
meta.data = data.frame(name = c("a_1", "a_2", "b_1", "b_2"), group = c("a", "a", "b", "b"))
rownames(meta.data) = meta.data[,1]
qbonedata = createQboneData(list1, meta.data, sampleid = 1)
qbone1 = createQboneObject(qbonedata, meta.data = meta.data)

## End(Not run)
```

defaultAssay

Get and set the default assay

Description

Get and set the default assay

Usage

```
defaultAssay(object, ...)

defaultAssay(object, ...) <- value

## S3 method for class 'Qbone'
defaultAssay(object, ...)

## S3 replacement method for class 'Qbone'
defaultAssay(object, ...) <- value
```


Arguments

object	An object
...	Arguments passed to other methods
value	Name of assay to set as default

Value

DefaultAssay: The name of the default assay
 defaultAssay<-: An object with the default assay updated

dxPlot	<i>Plot near-losslessness parameters.</i>
--------	---

Description

Plots the near-losslessness parameters minimum concordance (ρ^o) and average ($\bar{\rho}$) against the number of basis coefficients function of K_c in the reduced set.

Usage

```
dxPlot(object, sparsity = 0.001, ...)
```

Arguments

object	A Qboneobject
sparsity	Sparsity regularization parameter.
...	Arguments passed to other methods

dxPlotRev	<i>Plot number of basis coefficients.</i>
-----------	---

Description

Reversed dxPlot useful when plotting with other basis like principal components. minimum concordance (ρ^o) and average ($\bar{\rho}$) for quantlets basis, varying with the number of basis coefficients.

Usage

```
dxPlotRev(object, sparsity = 0.001, ...)
```

Arguments

object	A Qboneobject#'
sparsity	Sparsity regularization parameter.
...	Arguments passed to other methods

ecQuantlets

Get the Compute Empirical Coefficients for quantlets

Description

Compute empirical coefficients for quantlets

Usage

```
ecQuantlets(
  object,
  new.assay.name = "Empirical.Coefficients",
  data.assay = defaultAssay(object),
  k = NULL,
  sparsity = 0.001,
  ...
)
```

Arguments

object	A Qboneobject
new.assay.name	New assay name assigned to the quantlets data
data.assay	It is the name of the assay whose data will be used to compute the lasso list. Default is the data from the defaultAssay(object).
k	number of basis coefficients to keep. This is based on the number of dictionary elements (C) to keep. Default will pick the one chosen from dxPlot().
sparsity	Sparsity regularization parameter.
...	Arguments passed to other methods

eigenmm

Faster Matrix Multiplication

Description

Faster matrix multiplication using C++ Eigen. $A * B$
 Faster matrix multiplication using C++ Eigen. $t(A)$
 Faster matrix multiplication using C++ Eigen::Map $t(A)$
 Faster matrix multiplication using C++ Eigen::Map. $A * B$
 Faster matrix multiplication using C++ Eigen::Map. $A * t(B)$
 Faster matrix multiplication using C++ Eigen::Map. $t(A) * B$

Usage

```
eigenmm(A, B)

eigenmt(A)

eigenmapmt(A)

eigenmapmm(A, B)

eigenmapmmt(A, B)

eigenmapmtm(A, B)
```

Arguments

A	Matrices.
A, B	Matrices.

histogram3D	<i>Histogram Plot for each sample in 3D layout</i>
-------------	--

Description

Histogram Plot for each sample in 3D layout

Usage

```
histogram3D(
  object,
  title = "Histogram",
  binbreaks = 1000,
  data.assay = defaultAssay(object),
  plotting = "scatter3d",
  ...
)
```

Arguments

object	A Qboneobject
title	Title of the plot.
binbreaks	bin width breaks for the Histogram. Defulat is 1000
data.assay	It is the name of the assay whose data will be plotted
plotting	Plotting type, default is "scatter3d".
...	Arguments passed to other methods

idents

*Get, set, and manipulate an object's identity classes***Description**

Get, set, and manipulate an object's identity classes

Usage

```
idents(object, ...)

idents(object, ...) <- value

## S3 method for class 'Qbone'
idents(object, ...)

## S3 replacement method for class 'Qbone'
idents(object, samples = NULL, drop = FALSE, ...) <- value

## S3 method for class 'Qbone'
levels(x)

## S3 replacement method for class 'Qbone'
levels(x) <- value

## S3 method for class 'Qbone'
droplevels(x, ...)
```

Arguments

object	An Qbone object
...	Arguments passed to other methods
value	The name of the identities to pull from object metadata or the identities themselves
samples	Set cell identities for specific samples
drop	Drop unused levels

Value

idents<=: object with the cell identities changed

Examples

```
## Not run:
# Get sample identity classes
idents(qbone1)
```

```
## End(Not run)

## Not run:
# Set sample identity classes
# Can be used to set identities for specific samples to a new level
idents(qbone1, samples = 3:4) <- 'c'
idents(qbone1)

# Can also set idents from a value in object metadata
colnames(qbone1[[ ]])
idents(qbone1) <- colnames(qbone1[[ ]])[3]
# or
idents(qbone1) <- 'group2'
idents(qbone1)

## End(Not run)

## Not run:
# Get the levels of identity classes of a Qbone object
levels(x = qbone1)

## End(Not run)

## Not run:
# Reorder identity classes
levels(x = qbone1)
levels(x = qbone1) <- c('A', 'B')
levels(x = qbone1)

## End(Not run)
```

lassoList	<i>Use penalized regression (lasso) to find a sparse subset of dictionary elements</i>
-----------	--

Description

First construct overcomplete dictionary (Beta CDF). Then uses penalized regression (lasso) to find a sparse subset of dictionary elements.

Usage

```
lassoList(
  object,
  verbose = TRUE,
  new.assay.name = "Lasso.list",
  data.assay = defaultAssay(object),
  alpha = c(seq(0.1, 1, by = 0.1), seq(2, 100, by = 1)),
```

```

    beta = c(seq(0.1, 1, by = 0.1), seq(2, 100, by = 1)),
    assay.seed = .Random.seed,
    parallel = T,
    ...
)

```

Arguments

object	A Qboneobject
verbose	Print a progress bar
new.assay.name	New assay name assigned to the lassolist data
data.assay	It is the name of the assay whose data will be used to compute the lasso list. Default is the data from the defaultAssay(object).
alpha	Vector containing sequence of beta parameter for internal function generateBetaCDF()
beta	Vector containing sequence of beta parameter for internal function generateBetaCDF()
assay.seed	assay information to add into the QboneData object scale.data. The default of lassolist() will save the random seed for the run. Use .Random.seed <-object@assays[["Lasso.list"]][@scale.data[["lassolist"]] before run lassolist() for the same results.
parallel	If TRUE, use parallel foreach to fit each fold in cv.glmnet(). Default use registerDoMC() to register parallel. There is another function lassolist_parallel() for overall parallel computing for all sample.
...	Arguments passed to other methods

pdPlot

Predicted Density Plot

Description

Plot predicted densities

Usage

```

pdPlot(
  object,
  plot.col,
  group.names,
  mean.diff = F,
  var.diff = F,
  skewed.diff = F,
  kurtosis.diff = F,
  ...
)

```

Arguments

object	A Qboneobject
plot.col	Columns to plot from qfrModel() results object@assays[["Q.F.Regression"]][@scale.data[["mcm"]]. Corresponding to X1 (new covariates) agreement in qfrModel().
group.names	Group name for plot legend for argument plot.col.
mean.diff	T or F to add mean difference testing for two consecutive subjects posterior probability scores
var.diff	T or F to add variance difference testing for two consecutive subjects posterior probability scores
skewed.diff	T or F to add skewness difference testing for two consecutive subjects posterior probability scores
kurtosis.diff	T or F to add kurtosis difference testing for two consecutive subjects posterior probability scores
...	Arguments passed to other methods

preQuantlets	<i>Get the pre-quantlets basis functions</i>
--------------	--

Description

Take union set and find near-lossless subset as confirmed by cross-validated concordance correlation coefficient (CVCCC)

Usage

```
preQuantlets(
  object,
  new.assay.name = "Pre.Quantiles",
  data.assay = defaultAssay(object),
  p = signif(seq(0.001, 0.999, length = 1024), 4),
  alpha = c(seq(0.1, 1, by = 0.1), seq(2, 100, by = 1)),
  beta = c(seq(0.1, 1, by = 0.1), seq(2, 100, by = 1)),
  ...
)
```

Arguments

object	A Qboneobject
new.assay.name	New assay name assigned to the quantlets data
data.assay	It is the name of the assay whose data will be used to compute the lasso list. Default is the data from the defaultAssay(object).
p	Vector of length P in (0,1) Probability grids.
alpha	Vector containing sequence of beta parameter for internal function generateBetaCDF()
beta	Vector containing sequence of beta parameter for internal function generateBetaCDF()
...	Arguments passed to other methods

qbasisPlot	<i>Plot first n quantlet basis functions</i>
------------	--

Description

Plot first n quantlet basis functions

Usage

```
qbasisPlot(object, n = 16, ...)
```

Arguments

object	A Qboneobject
n	Number of first n basis functions to plot, default = 16.
...	Arguments passed to other methods

qbasisPlot3D	<i>3D Plot of first n quantlet basis functions</i>
--------------	--

Description

Plot first n quantlet basis functions in 3D Plots

Usage

```
qbasisPlot3D(object, n = 16, ...)
```

Arguments

object	A Qboneobject
n	Number of first n basis functions to plot, default = 16.
...	Arguments passed to other methods

Qbone-class

*The Qbone Class***Description**

The Qbone object is a representation of data analysis using Quantile Functional Regression using Quantlets (doi: 10.1080/01621459.2019.1609969) for R.

Slots

assays A list of QboneData Object for this project.
 meta.data Meta information regarding each sample.
 active.assay Name of the active, or default, assay; settable using [defaultAssay](#)
 active.ident The active identity for this Qbone object; settable using [idents](#)
 project.name Name of the project
 version Version of Qbone this object was built under

Qbone-methods

*Qbone Methods***Description**

Methods for [Qbone](#) objects for generics defined in other packages

Usage

```
## S3 method for class 'Qbone'
x[[i, ..., drop = FALSE]]

## S4 replacement method for signature 'Qbone'
x[[i, j, ...]] <- value
```

Arguments

x, object	A Qbone object
i	Depends on the method [, [[<- Name of one or more metadata columns or an associated object; associated objects include QboneData double check
...	Arguments passed to other methods
drop	See drop
j, samples	Sample names or indices
value	Additional metadata or associated objects to add; note : can pass NULL to remove metadata or an associated object
n	The number of rows of metadata to return

Value

`[[`: If `i` is missing, the metadata data frame; if `i` is a vector of metadata names, a data frame with the requested metadata, otherwise, the requested associated object

`[[<-`: `x` with the metadata or associated objects added as `i`; if value is `NULL`, removes metadata or associated object `i` from object `x`

Functions

- `[[`: Metadata and associated object accessor
- ``[[` (Qbone) <-value`: Metadata and associated object accessor

QboneData-class	<i>The QboneData Class The QboneData object is the basic unit of Qbone.</i>
-----------------	---

Description

The QboneData Class The QboneData object is the basic unit of Qbone.

Slots

`data` Sample data

`scale.data` Parameter of data processing

`assay.name` name of assay

`assay.orig` Original assay that this assay is based off of. Used to track assay provenance

`meta.assays` Metadata for the data processing

See Also

[QboneData-methods](#)

QboneData-methods	<i>QboneData Methods QboneData Methods</i>
-------------------	--

Description

Methods for [QboneData](#) objects for generics defined in other packages

Usage

```
## S3 method for class 'QboneData'
x[[i, ..., drop = FALSE]]
```

Arguments

x, object	An QboneData object
i, features	For <code>[[</code> : metadata names; for all other methods, feature names or indices
...	Arguments passed to other methods
drop	See drop
j, samples	sample names or indices

Value

`[[`: If `i` is missing, the metadata data frame; if `i` is a vector of metadata names, a data frame with the requested metadata, otherwise, the requested associated object

Functions

- `[[`: Metadata and associated object accessor

qfrModel	<i>Fit the quantile functional regression model</i>
----------	---

Description

Fit the quantlet space model in using Markov chain Monte Carlo (MCMC)

Usage

```
qfrModel(
  object,
  new.assay.name = "Q.F.Reggression",
  data.assay = defaultAssay(object),
  X = NULL,
  X1 = NULL,
  delta2 = 0.95,
  H = NULL,
  pct.range = c(0.05, 0.95),
  assay.seed2 = .Random.seed,
  ...
)
```

Arguments

object	A Qboneobject
new.assay.name	New assay name assigned to the quantlets data
data.assay	It is the name of the assay whose data will be used to compute the lasso list. Default is the data from the <code>defaultAssay(object)</code> .
X	Covariates (N by A matrix, N is the number of observations)

X1	New covariates for inferencing MCMC fit result.
delta2	Cutoff percentage of the energy for the functional coefficients
H	Number of clustering groups. Cluster basis indices based on their eigen-values.
pct.range	Percentage range from original data set. Use as the range of the domain for the density estimate. Default is c(0.05, 0.95). You can override it by assign xrange = c(min,max)
...	Arguments passed to other methods

quantileFPlot3D	<i>3D Plot of first n quantlet with observed and predicted quantile function</i>
-----------------	--

Description

3D Plot of first n quantlet with observed and predicted quantile function

Usage

```
quantileFPlot3D(
  object,
  n = 16,
  group = NULL,
  data.assay = defaultAssay(object),
  plot = "All",
  ...
)
```

Arguments

object	A Qboneobject
n	Number of first n basis functions to plot, default = 16.
group	Group to be plotted. All group will be plotted if NULL. Default is NULL.
data.assay	It is the name of the assay whose data will be plotted
plot	Plot "Observed", "Quantlets", "Predicted" or "All" which is "Observed -> Quantlets -> Predicted". Default is "All"
...	Arguments passed to other methods

readQbone	<i>Read Qbone from files</i>
-----------	------------------------------

Description

Read Qbone from files

Usage

```
readQbone(data.dir, groupbyfolder = F, data.column = 1)
```

Arguments

data.dir	the folder directory where the raw csv file saved
groupbyfolder	Use parent folder names as sample group information if set to T, default is F
data.column	column number of sample data in the raw csv file

samples	<i>Get samples present in an object</i>
---------	---

Description

Get samples present in an object

Usage

```
samples(x)

## S3 method for class 'QboneData'
samples(x)

## S3 method for class 'Qbone'
samples(x)
```

Arguments

x	An object
---	-----------

Value

A vector of sample names

Examples

```
## Not run:
samples(x = qbone1)

## End(Not run)
```

set-if-null

Set a default value depending on if an object is NULL From rlang package.

Description

Set a default value depending on if an object is NULL From rlang package.

Usage

```
x %||% y
```

Arguments

x	An object to test
y	A default value

Value

For %||%: y if x is NULL otherwise x

Examples

```
1 %||% 2
NULL %||% 2
```

thinData

Thin the data If the data set is too big and the calculation is too heavy for your computer you may use this function to thin the data to make the calculation faster. Here we use thin the data proportionally on the sorted data. The new data set should reserve the same quantile feature as it was sequentially subsetted from ordered data set.

Description

Thin the data If the data set is too big and the calculation is too heavy for your computer you may use this function to thin the data to make the calculation faster. Here we use thin the data proportionally on the sorted data. The new data set should reserve the same quantile feature as it was sequentially subsetted from ordered data set.

Usage

```
thinData(object, new.assay.name = "Thin", prop = NULL)
```

Arguments

<code>object</code>	An Qboneobject.
<code>new.assay.name</code>	new assay name assigned to the thined data
<code>prop</code>	proportion to keep from the original data. By default the scale.data of <code>thinData()</code> will save the prop of thinning

Index

- * **Data thin**
 - thinData, [22](#)
- * **QboneData**
 - QboneData-methods, [18](#)
- * **data-access**
 - assayData, [4](#)
 - assays, [5](#)
 - [[.Qbone (Qbone-methods), [17](#)
 - [[.QboneData (QboneData-methods), [18](#)
 - [[<-, Qbone-method (Qbone-methods), [17](#)
 - _PACKAGE (Qbone-package), [3](#)
- addMetaData, [3](#)
- assay, [5](#)
- assayData, [4](#)
- assays, [5](#)
- createQboneData, [5](#)
- createQboneObject, [6](#)
- default assay, [4](#)
- defaultAssay, [8](#), [17](#)
- defaultAssay<- (defaultAssay), [8](#)
- drop, [17](#), [19](#)
- droplevels.Qbone (idents), [12](#)
- dxPlot, [9](#)
- dxPlotRev, [9](#)
- ecQuantlets, [10](#)
- eigenmapmm (eigenmm), [10](#)
- eigenmapmmt (eigenmm), [10](#)
- eigenmapmt (eigenmm), [10](#)
- eigenmapmtm (eigenmm), [10](#)
- eigenmm, [10](#)
- eigenmt (eigenmm), [10](#)
- getQboneData (assayData), [4](#)
- histogram3D, [11](#)
- idents, [12](#), [17](#)
- idents<- (idents), [12](#)
- lassoList, [13](#)
- levels.Qbone (idents), [12](#)
- levels<- .Qbone (idents), [12](#)
- pdPlot, [14](#)
- preQuantlets, [15](#)
- project, [7](#)
- qbasisPlot, [16](#)
- qbasisPlot3D, [16](#)
- Qbone, [5](#), [17](#)
- Qbone (Qbone-class), [17](#)
- Qbone-class, [17](#)
- Qbone-methods, [17](#)
- Qbone-package, [3](#)
- QboneData, [4](#), [6](#), [17–19](#)
- QboneData (QboneData-class), [18](#)
- QboneData-class, [18](#)
- QboneData-methods, [18](#)
- qfrModel, [19](#)
- quantileFPlot3D, [20](#)
- readQbone, [21](#)
- samples, [21](#)
- set-if-null, [22](#)
- thinData, [22](#)