

# Package ‘regioneReloaded’

January 23, 2026

**Type** Package

**Title** RegioneReloaded: Multiple Association for Genomic Region Sets

**Version** 1.12.0

**URL** <https://github.com/RMalinverni/regioneReload>

**Description** RegioneReloaded is a package that allows simultaneous analysis of associations between genomic region sets, enabling clustering of data and the creation of ready-to-publish graphs. It takes over and expands on all the features of its predecessor regioneR. It also incorporates a strategy to improve p-value calculations and normalize z-scores coming from multiple analysis to allow for their direct comparison. RegioneReloaded builds upon regioneR by adding new plotting functions for obtaining publication-ready graphs.

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.2.1

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 4.2), regioneR

**Imports** stats, RColorBrewer, Rtsne, umap, ggplot2, ggrepel, reshape2, methods, scales, cluster, grid, grDevices

**Suggests** rmarkdown, BiocStyle, GenomeInfoDb, knitr, testthat (>= 3.0.0)

**biocViews** Genetics, ChIPSeq, DNASeq, MethylSeq, CopyNumberVariation, Clustering, MultipleComparison

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/regioneReloaded>

**git\_branch** RELEASE\_3\_22

**git\_last\_commit** f9a9462

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.22

**Date/Publication** 2026-01-22

**Author** Roberto Malinverni [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-0113-3417>>),  
David Corujo [aut],  
Bernat Gel [aut]

**Maintainer** Roberto Malinverni <[roberto.malinverni@gmail.com](mailto:roberto.malinverni@gmail.com)>

## Contents

|                                     |    |
|-------------------------------------|----|
| AlienGenome . . . . .               | 3  |
| AlienRSList_broad . . . . .         | 3  |
| AlienRSList_narrow . . . . .        | 4  |
| chooseHclustMet . . . . .           | 4  |
| cleanCrosswiseMatrix . . . . .      | 5  |
| createUniverse . . . . .            | 6  |
| crosswiseMatrix . . . . .           | 6  |
| crosswisePermTest . . . . .         | 7  |
| cw_Alien . . . . .                  | 8  |
| cw_Alien_RaR . . . . .              | 9  |
| cw_Alien_ReG . . . . .              | 9  |
| cw_Alien_ReG_no_Square . . . . .    | 9  |
| cw_Alien_ReR . . . . .              | 10 |
| DFfromLZ . . . . .                  | 10 |
| genoMatriXeR-class . . . . .        | 11 |
| getHClust . . . . .                 | 11 |
| getMatrix . . . . .                 | 12 |
| getMultiEvaluation . . . . .        | 13 |
| getParameters . . . . .             | 13 |
| gmxrMatrix . . . . .                | 14 |
| gmxrMatrix<- . . . . .              | 15 |
| gmxrMultiOverlaps . . . . .         | 15 |
| gmxrParam . . . . .                 | 16 |
| makeCrosswiseMatrix . . . . .       | 16 |
| makeLZMatrix . . . . .              | 18 |
| mlzsMatrix . . . . .                | 19 |
| mlzsMatrix<- . . . . .              | 19 |
| mlzsMultiLocalZscores . . . . .     | 20 |
| mlzsParam . . . . .                 | 20 |
| mlzsParam<- . . . . .               | 21 |
| mLZ_regA_ReG . . . . .              | 21 |
| mLZ_regA_ReG_br . . . . .           | 21 |
| mLZ_regD_ReG . . . . .              | 22 |
| multiLocalZscore . . . . .          | 22 |
| multiLocalZScore-class . . . . .    | 24 |
| multiPermTest . . . . .             | 24 |
| plotCrosswiseDimRed . . . . .       | 25 |
| plotCrosswiseMatrix . . . . .       | 27 |
| plotLocalZScoreMatrix . . . . .     | 28 |
| plotSingleLZ . . . . .              | 29 |
| plotSinglePT . . . . .              | 30 |
| print,genoMatriXeR-method . . . . . | 31 |
| randomizeRegionsPerc . . . . .      | 32 |
| similarRegionSet . . . . .          | 33 |
| subList . . . . .                   | 34 |

---

*AlienGenome*

---

*AlienGenome*

---

**Description**

The Alien Genome is an artificial genomic coordinates system for the purposes of testing and demonstrating the functions of `regioneReload` with a low computing time.

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class [GRanges](#).

**Details**

The Alien Genome consists of four chromosomes and is generated by the following code:

```
AlienGenome <-  
toGRanges(data.frame(  
  chr = c("A1Chr1", "A1Chr2", "A1Chr3", "A1Chr4"),  
  start = c(rep(1, 4)),  
  end = c(2e6, 1e6, 5e5, 1e5)  
))
```

---

*AlienRSList\_broad*

---

*AlienRSList\_broad*

---

**Description**

List of region sets (as [GRanges](#)) on the [AlienGenome](#).

**Usage**

```
data(cw_Alien)
```

**Format**

A list of [GRanges](#) objects.

**Details**

This region sets are generated for the purpose of demonstrating the functions of `RegionReloaded` with a low computing time and "predictable" associations. The regions are generated with by combining [createRandomRegions\(\)](#) and [similarRegionSet\(\)](#) so that there is a known overlap between certain region sets. To see a full description of this sample data and the code used to generate it, see the `RegionReloaded` vignette.

---

|                    |                           |
|--------------------|---------------------------|
| AlienRSList_narrow | <i>AlienRSList_narrow</i> |
|--------------------|---------------------------|

---

### Description

List of region sets (as [GRanges](#)) on the [AlienGenome](#).

### Usage

```
data(cw_Alien)
```

### Format

A list of [GRanges](#) objects.

### Details

This region sets are generated for the purpose of demonstrating the functions of RegioneReloaded with a low computing time and "predictable" associations. The regions are generated with by combining [createRandomRegions\(\)](#) and [similarRegionSet\(\)](#) so that there is a known overlap between certain region sets. To see a full description of this sample data and the code used to generate it, see the RegioneReloaded vignette.

---

|                 |                        |
|-----------------|------------------------|
| chooseHclustMet | <i>chooseHclustMet</i> |
|-----------------|------------------------|

---

### Description

Evaluate and choose the best method for clustering a matrix using the [hclust\(\)](#) function.

### Usage

```
chooseHclustMet(GM, scale = FALSE, vecMet = NULL, distHC = "euclidean")
```

### Arguments

|        |                                                                                                                                                                                                                    |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GM     | matrix, numerical matrix.                                                                                                                                                                                          |
| scale  | logical, if TRUE, the clustering will be performed using the scaled matrix. (default = FALSE)                                                                                                                      |
| vecMet | character, vector of methods that will be tested in the function. If NULL, the following methods will be tested: "complete", "average", "single", "ward.D2", "median", "centroid" and "mcquitty". (default = NULL) |
| distHC | character, the distance measure to be used from those available in <a href="#">dist()</a> . (default = "euclidean")                                                                                                |

### Value

An object of class [hclust](#)

**See Also**[hclust\(\)](#)**Examples**

```
M1 <- matrix(1:18, nrow = 6, ncol = 3)
set.seed(42)
M2 <- matrix(sample(100, 18), nrow = 6, ncol = 3)
GM <- cbind(M1, M2)

chooseHclustMet(GM)
```

---

|                      |                             |
|----------------------|-----------------------------|
| cleanCrosswiseMatrix | <i>cleanCrosswiseMatrix</i> |
|----------------------|-----------------------------|

---

**Description**

Clean and scale a matrix from a genoMatriXeR object

**Usage**

```
cleanCrosswiseMatrix(GM, GM_pv, pvcut, scale, subEX)
```

**Arguments**

|       |                                                                                                                                                                                                           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GM    | matrix, numerical matrix of z-scores.                                                                                                                                                                     |
| GM_pv | matrix, numerical matrix of pvalues.                                                                                                                                                                      |
| pvcut | numeric, the z-score value is substituted by subEX (0 by default) for all the associations with an adj.pvalue (as calculated in <a href="#">crosswisePermTest()</a> ) higher than pvcut. (default = 0.05) |
| scale | logical, if TRUE the matrix will be scaled. (default = FALSE)                                                                                                                                             |
| subEX | numeric, value used to substitute the z-score values when the associated pvalue is higher than pvcut. (default = 0)                                                                                       |

**Value**

a matrix filtered for a matrix of pvalue

**See Also**[makeCrosswiseMatrix\(\)](#)

---

|                |                       |
|----------------|-----------------------|
| createUniverse | <i>createUniverse</i> |
|----------------|-----------------------|

---

### Description

Create the universe parameter for `regioneR::resampleRegions()` using all unique regions present in Alist.

### Usage

```
createUniverse(Alist, joinR = TRUE)
```

### Arguments

|       |                                                                                                                             |
|-------|-----------------------------------------------------------------------------------------------------------------------------|
| Alist | list of regions set in a format accepted for <code>regioneR</code>                                                          |
| joinR | logical, if TRUE all the regions will be joined using the function <code>regioneR::joinRegions()</code> . (default == TRUE) |

### Value

A list of `GRanges` objects

### Examples

```
data("cw_Alien")

universe <- createUniverse(AlienRSList_narrow)
```

---

|                 |                        |
|-----------------|------------------------|
| crosswiseMatrix | <i>crosswiseMatrix</i> |
|-----------------|------------------------|

---

### Description

Create a matrix from a `genoMatriXeR` object.

### Usage

```
crosswiseMatrix(mPT, zs.type='norm_zscore',...)
```

### Arguments

|         |                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| mPT     | an object of class <code>genoMatriXeR</code> .                                                                                                         |
| zs.type | character, z-score type to use to generate the matrix, either raw z-score ("zs-core") or normalized z-score ("norm_zscore"). (default = "norm_zscore") |
| ...     | further arguments to be passed to other methods.                                                                                                       |

### Value

a numeric matrix

---

|                   |                          |
|-------------------|--------------------------|
| crosswisePermTest | <i>crosswisePermTest</i> |
|-------------------|--------------------------|

---

## Description

Perform multiple permutation tests between each element in two lists of region sets.

## Usage

```
crosswisePermTest(Alist, Blist = NULL, sampling = FALSE, fraction = 0.15,
  min_sampling = 5000, ranFUN = "randomizeRegions", evFUN = "numOverlaps",
  ntimes = 100, universe = NULL, adj_pv_method = "BH",
  genome = "hg19", ...)
```

## Arguments

|               |                                                                                                                                                                                                                                                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alist, Blist  | <a href="#">GRangesList</a> or list of region sets in any accepted formats by <a href="#">regioneR</a> package ( <a href="#">GRanges</a> , <a href="#">data.frame</a> etc.).                                                                                              |
| sampling      | logical, if TRUE the function will use only a sample of each element of Alist to perform the test as specified in fraction. (default = FALSE)                                                                                                                             |
| fraction      | logical, if sampling=TRUE, defines the fraction of the region sets used to perform the test. (default = 0.15)                                                                                                                                                             |
| min_sampling  | numeric, minimum number of regions accepted after sampling is performed with the specified fraction. If the number of sampled regions is less than min_sampling, the number specified by min_sampling will be used as number of regions sampled instead. (default = 5000) |
| ranFUN        | character, the randomization strategy used for the test, see <a href="#">regioneR</a> . (default = "randomizeRegions")                                                                                                                                                    |
| evFUN         | character, the evaluation strategy used for the test, see <a href="#">regioneR</a> . (default = "numOverlaps")                                                                                                                                                            |
| ntimes        | numeric, number of permutations used in the test. (default = 100)                                                                                                                                                                                                         |
| universe      | region set to use as universe, used only when <a href="#">regioneR::resampleRegions()</a> function is selected. (default = NULL)                                                                                                                                          |
| adj_pv_method | character, the method used for the calculation of the adjusted p-value, to choose between the options of <a href="#">p.adjust()</a> . (default = "BH")                                                                                                                    |
| genome        | character or <a href="#">GRanges</a> , genome used to compute the randomization. (default = "hg19")                                                                                                                                                                       |
| ...           | further arguments to be passed to other methods.                                                                                                                                                                                                                          |

## Details

This function performs multiple permutation tests for all pairwise combinations of the elements in two lists of region sets. Essentially, it uses the [regioneR::permTest\(\)](#) function and its associated randomization and evaluation functions. It creates and returns a [genoMatrixeR](#) object with the result of the permutation tests stored in the multiOverlaps slot. In addition, all the parameters used for the test are stored in the parameters slot.

**Value**

A object of class `genoMatriXeR` containing three slots

- `@parameters`
- `@multioverlaps`
- `@matrix`

**See Also**

`genoMatriXeR`, `regioneR`, `regioneR::permTest()`, `regioneR::overlapPermTest()`

**Examples**

```
fakeGenome <- regioneR::toGRanges("chrF", 1, 1000)
regA <- regioneR::createRandomRegions(nregions = 10, length.mean = 10,
length.sd = 2, genome = fakeGenome)
regB <- regioneR::createRandomRegions(nregions = 10, length.mean = 10,
length.sd = 2, genome = fakeGenome)
regAs <- similarRegionSet(GR = regA, genome = fakeGenome, name = "A",
vectorPerc = seq(0.1, 0.3, by = 0.1))
regBs <- similarRegionSet(GR = regB, genome = fakeGenome, name = "B",
vectorPerc = seq(0.1, 0.3, by = 0.1))
ABList <- c(regAs, regBs)
cw_ptAB <- crosswisePermTest(ABList, genome = fakeGenome, ntimes = 10)
print(cw_ptAB)
```

---

cw\_Alien

cw\_Alien

---

**Description**

Alien Genome crosswise matrix using `regioneR::randomizeRegions`, `regioneR::circularRandomizeRegions`, `regioneR::resampleRegions`, `regioneR::resampleGenome` functions as permutation strategies.

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class `genoMatriXeR`; see `makeCrosswiseMatrix()`.

---

|              |                     |
|--------------|---------------------|
| cw_Alien_RaR | <i>cw_Alien_RaR</i> |
|--------------|---------------------|

---

**Description**

Alien Genome crosswise matrix using [regioneR::randomizeRegions\(\)](#) function a permutation strategy. Alist = AlienRSList\_narrow, Blist = AlienRSList\_narrow

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class [genoMatrixeR](#); see [makeCrosswiseMatrix\(\)](#).

---

|              |                     |
|--------------|---------------------|
| cw_Alien_ReG | <i>cw_Alien_ReG</i> |
|--------------|---------------------|

---

**Description**

Alien Genome crosswise matrix using [regioneR::resampleGenome\(\)](#) function as permutations strategy. Alist = AlienRSList\_narrow, Blist = AlienRSList\_narrow

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class [genoMatrixeR](#); see [makeCrosswiseMatrix\(\)](#).

---

|                        |                               |
|------------------------|-------------------------------|
| cw_Alien_ReG_no_Square | <i>cw_Alien_ReG_no_Square</i> |
|------------------------|-------------------------------|

---

**Description**

Alien Genome crosswise matrix using [regioneR::resampleGenome\(\)](#) function as permutations strategy. Alist = AlienRSList\_narrow, Blist = AlienRSList\_broad

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class [genoMatrixeR](#); see [makeCrosswiseMatrix\(\)](#).

---

|              |                     |
|--------------|---------------------|
| cw_Alien_ReR | <i>cw_Alien_ReR</i> |
|--------------|---------------------|

---

**Description**

Alien Genome crosswise matrix using `regioneR::resampleRegions()` function a permutation strategy. Alist = AlienRSList\_narrow, Blist = AlienRSList\_narrow

**Usage**

`data(cw_Alien)`

**Format**

An objects of class `genoMatriXeR`; see `makeCrosswiseMatrix()`.

---

|          |                                               |
|----------|-----------------------------------------------|
| DFfromLZ | <i>Dataframe from MultiLocalZScore object</i> |
|----------|-----------------------------------------------|

---

**Description**

Dataframe from MultiLocalZScore object

**Usage**

`DFfromLZ( mLZ, RS )`

**Arguments**

- mLZ                      object class MultilocalZScore
- RS                        string name of single region set present in the mLZ object

**Value**

a data.frame from `multiLocalZScore`

---

|                    |                           |
|--------------------|---------------------------|
| genoMatriXeR-class | <i>genoMatriXeR Class</i> |
|--------------------|---------------------------|

---

### Description

An S4 class for "genoMatriXeR" object.

### Slots

**parameters** List of parameters used to create the object.

**multiOverlaps** Results of multiple pairwise permutation tests generated with [crosswisePermTest\(\)](#).

**matrix** List of numerical matrices containing z-score, pvalues and correlation values generated with [makeCrosswiseMatrix\(\)](#)

### Examples

```
data("cw_Alien")

AlienRSList_narrow_small <- AlienRSList_narrow[c("regA","regB","regC")]

cw_test <- crosswisePermTest(Alist = AlienRSList_narrow_small,Blist = AlienRSList_narrow_small,
                             sampling = FALSE, genome = AlienGenome, per.chromosome = TRUE,
                             ranFUN = "resampleGenome", evFUN = "numOverlaps",
                             ntimes = 10, mc.cores = 2)

class(cw_test)
```

---

|           |                  |
|-----------|------------------|
| getHClust | <i>getHClust</i> |
|-----------|------------------|

---

### Description

get Object of class [hclust](#) from [genoMatriXeR](#) or [multiLocalZScore](#)

### Usage

```
getHClust( rR, hctype = "rows")
```

### Arguments

|        |                                                       |
|--------|-------------------------------------------------------|
| rR     | A genoMatriXeR or multiLocalZScore object.            |
| hctype | character. Can be "rows" or "cols". (default= "cols") |

### Value

an object of class [hclust](#)

### See Also

[genoMatriXeR](#), [multiLocalZScore](#), [hclust](#)

**Examples**

```
data("cw_Alien")

cw_Alien_ReG <- makeCrosswiseMatrix(cw_Alien_ReG)
hc <- getHClust(cw_Alien_ReG)

plot(hc)
```

---

**getMatrix***Get Matrix*

---

**Description**

Returns the matrix from an [genoMatrixeR](#) or [multiLocalZScore](#) object.

**Usage**

```
getMatrix(rR)
```

**Arguments**

rR                      [genoMatrixeR](#) or [multiLocalZScore](#) object

**Value**

a numerical matrix from a

**See Also**

[genoMatrixeR](#), [multiLocalZScore](#), [makeCrosswiseMatrix](#), [makeLZMatrix](#)

**Examples**

```
data("cw_Alien")

cw_Alien_ReG <- makeCrosswiseMatrix(cw_Alien_ReG)
mtx <- getMatrix(cw_Alien_ReG)

mtx

data("cw_Alien")

cw_Alien_RaR <- makeCrosswiseMatrix(cw_Alien_RaR)
GM <- getMatrix(cw_Alien_RaR)

GM
```

---

|                    |                           |
|--------------------|---------------------------|
| getMultiEvaluation | <i>getMultiEvaluation</i> |
|--------------------|---------------------------|

---

### Description

Get multiEvaluation slot from [genoMatriXeR](#) or [multiLocalZScore](#) class.

### Usage

```
getMultiEvaluation( rR, namesRS = NULL)
```

### Arguments

|         |                                            |
|---------|--------------------------------------------|
| rR      | A genoMatriXeR or multiLocalZScore object. |
| namesRS | a vector of names. (default = NA)          |

### Value

If rR is a [genoMatriXeR](#) object, a list of data frames resuming the associations results. If rR is a [multiLocalZScore](#) object, a list of two elements: "resumeTable" that is a data frame summarizing the associations and "shifts", a list of shifts computed from [multiLocalZscore\(\)](#) function for the elements indicated in the nameRS vector.

### See Also

[genoMatriXeR](#), [multiLocalZScore](#)

### Examples

```
data("cw_Alien")

mevs <- getMultiEvaluation(cw_Alien_ReG, names = "regA")

mevs
```

---

|               |                      |
|---------------|----------------------|
| getParameters | <i>getParameters</i> |
|---------------|----------------------|

---

### Description

Get parameters from a genoMatriXeR or multiLocalZScore class object.

### Usage

```
getParameters(rR, show_err = FALSE)
```

**Arguments**

`rR` A `genoMatriXeR` or `multiLocalZScore` class object.

`show_err` logical, if TRUE the function returns a list with two dataframes: one containing the parameter values and one with any error messages that have been generated during the permutation test iterations when running [crosswisePermTest](#).

**Value**

A dataframe with parameters and values, or a list with two dataframes with parameters and errors information.

**See Also**

[genoMatriXeR](#), [multiLocalZScore](#)

**Examples**

```
data("cw_Alien")

prm <- getParameters(cw_Alien_ReG)

prm
```

---

`gmxrMatrix`

*Export matrix slot from genoMatriXeR objects*

---

**Description**

Export matrix slot from `genoMatriXeR` objects

**Usage**

```
gmxrMatrix(x)
```

**Arguments**

`x` A `genoMatriXeR` object.

**Value**

matrix slot of a `genoMatriXeR` object.

**See Also**

[genoMatriXeR](#)

---

|              |                                                       |
|--------------|-------------------------------------------------------|
| gmxrMatrix<- | <i>Set matrix slot value of a genoMatriXeR object</i> |
|--------------|-------------------------------------------------------|

---

**Description**

Set matrix slot value of a genoMatriXeR object

**Usage**

```
gmxrMatrix(x) <- value
```

**Arguments**

|       |                                     |
|-------|-------------------------------------|
| x     | A genoMatriXeR object.              |
| value | Value to assign to the matrix slot. |

**Value**

A genoMatriXeR object.

---

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| gmxrMultiOverlaps | <i>Export multiOverlaps slot from genoMatriXeR objects</i> |
|-------------------|------------------------------------------------------------|

---

**Description**

Export multiOverlaps slot from genoMatriXeR objects

**Usage**

```
gmxrMultiOverlaps(x)
```

**Arguments**

|   |                        |
|---|------------------------|
| x | A genoMatriXeR object. |
|---|------------------------|

**Value**

multiOverlaps slot of a genoMatriXeR.

**See Also**

[genoMatriXeR](#)

---

gmxrParam

Export parameters from genoMatriXeR objects

---

### Description

Export parameters from genoMatriXeR objects

### Usage

```
gmxrParam(x)
```

### Arguments

**x**                      A genoMatriXeR object.

### Value

Parameters slot of a genoMatriXeR object.

### See Also

[genoMatriXeR](#)

---

makeCrosswiseMatrix

*makeCrosswiseMatrix*


---

### Description

Populate the matrix slot in a [genoMatriXeR](#) object.

### Usage

```
makeCrosswiseMatrix(mPT, clusterize = TRUE, hc.method = NULL, dist.method = "euclidean",
  transform = FALSE, scale = FALSE, zs.type = 'norm_zscore', symm_matrix = TRUE,
  selectRow = NULL, selectCol = NULL, pvcut = 1, subEX = 0, GM_diag = TRUE, ...)
```

### Arguments

|                    |                                                                                                                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mPT</b>         | an object of class <a href="#">genoMatriXeR</a> .                                                                                                                                                                          |
| <b>clusterize</b>  | logical, if TRUE the matrix will be clustered using the method specified by <code>hc.method</code> (default = TRUE)                                                                                                        |
| <b>hc.method</b>   | character, select the <a href="#">hclust()</a> method to use for clustering the matrix. If NULL, the clustering method will be automatically selected by the function <a href="#">chooseHclustMet()</a> . (default = NULL) |
| <b>dist.method</b> | character, the distance measure to be used from those available in <a href="#">dist()</a> . (default = "euclidean")                                                                                                        |
| <b>transform</b>   | logical, if TRUE the matrix will be transformed using the function <a href="#">t()</a> . (default = FALSE)                                                                                                                 |
| <b>scale</b>       | logical, if TRUE the matrix will be scaled. (default = FALSE)                                                                                                                                                              |

|                      |                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| zs.type              | character, z-score type to use to generate the matrix, either raw z-score ("zs-core") or normalized z-score ("norm_zscore"). (default = "norm_zscore")                                                    |
| symm_matrix          | logical, if TRUE the matrix will be treated as symmetrical (same clustering for rows and columns). (default = TRUE)                                                                                       |
| selectRow, selectCol | vector, the matrix will be reduced selecting the rows and/or columns in this vector. (default = NULL)                                                                                                     |
| pvcut                | numeric, the z-score value is substituted by subEX (0 by default) for all the associations with an adj.pvalue (as calculated in <a href="#">crosswisePermTest()</a> ) higher than pvcut. (default = 0.05) |
| subEX                | numeric, value used to substitute the z-score values when the associated pvalue is higher than pvcut. (default = 0)                                                                                       |
| GM_diag              | logic, if FALSE the values of the diagonal will be set to 0. (default = TRUE)                                                                                                                             |
| ...                  | further arguments to be passed to other methods.                                                                                                                                                          |

## Details

This function will create a series of matrices of z-scores, adj.pvalues and pearson correlation values from all the pairwise permutation tests stored in the multiOverlaps slot of a [genoMatrixeR](#) as calculated with [multiPermTest\(\)](#). These matrices will then be stored in the matrix slot of the [genoMatrixeR](#) object. In addition, clustering will be performed on the association matrices using [hclust](#).

## Value

An object of class [genoMatrixeR](#) containing three slots, with a populated matrix slot.

- @parameters
- @multioverlaps
- @matrix

## See Also

[crosswisePermTest\(\)](#), [chooseHclustMet\(\)](#), [plotCrosswiseMatrix\(\)](#)

## Examples

```
data("cw_Alien")

cw_Alien_ReG <- makeCrosswiseMatrix(cw_Alien_ReG)

summary(cw_Alien_ReG)
```

makeLZMatrix

*Make Local Z-Score Matrix***Description**

Create a local z-score matrix from a [multiLocalZScore](#) object and save it in its `matrix` slot.

**Usage**

```
makeLZMatrix(mlZA, normalize = TRUE, clusterize = TRUE,
             centralize = NA, hc.method = NULL, dist.method = "euclidean",
             scale = FALSE, ...)
```

**Arguments**

|                          |                                                                                                                                                                                                                                                 |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mlZA</code>        | an object of class <a href="#">multiLocalZScore</a> or a numerical matrix.                                                                                                                                                                      |
| <code>normalize</code>   | logical, if TRUE the z-score values in the matrix will be normalized. (default = FALSE)                                                                                                                                                         |
| <code>clusterize</code>  | logical, if TRUE the matrix will be clustered using the method specified by <code>hc.method</code> (default = TRUE)                                                                                                                             |
| <code>centralize</code>  | numeric, only z-score values in a number of steps (defined by <code>centralize</code> ) around the center of the local association will be used for clustering. If NA, all the values in the matrix will be used for clustering. (default = NA) |
| <code>hc.method</code>   | character, select the <a href="#">hclust()</a> method to use for clustering the matrix. If NULL, the clustering method will be automatically selected by the function <a href="#">chooseHclustMet()</a> . (default = NULL)                      |
| <code>dist.method</code> | character, the distance measure to be used from those available in <a href="#">dist()</a> . (default = "euclidean")                                                                                                                             |
| <code>scale</code>       | logical, if TRUE the matrix will be scaled. (default = FALSE)                                                                                                                                                                                   |
| <code>...</code>         | further arguments to be passed to other methods.                                                                                                                                                                                                |

**Value**

A object of class [multiLocalZScore](#) containing three slots, with a populated `matrix` slot.

- `@parameters`
- `@multiLocalZscores`
- `@matrix`

**See Also**

[localZScore](#)

**Examples**

```
data("cw_Alien")
```

---

|            |                                                    |
|------------|----------------------------------------------------|
| mlzsMatrix | <i>Export matrix from multiLocalZScore objects</i> |
|------------|----------------------------------------------------|

---

**Description**

Export matrix from multiLocalZScore objects

**Usage**

```
mlzsMatrix(x)
```

**Arguments**

x                      A multiLocalZScore object.

**Value**

A printed output for multiLocalZScore matrix slot.

**See Also**

[multiLocalZScore](#)

---

|              |                                                            |
|--------------|------------------------------------------------------------|
| mlzsMatrix<- | <i>Set matrix slot value of a multiLocalZScore object.</i> |
|--------------|------------------------------------------------------------|

---

**Description**

Set matrix slot value of a multiLocalZScore object.

**Usage**

```
mlzsMatrix(x) <- value
```

**Arguments**

x                      A multiLocalZScore object.  
value                  Value to assign to the multiLocalZScore slot.

**Value**

A multiLocalZScore object.

---

|                       |                                                                    |
|-----------------------|--------------------------------------------------------------------|
| mlzsMultiLocalZscores | <i>Export multiLocalZscores slot from multiLocalZScore objects</i> |
|-----------------------|--------------------------------------------------------------------|

---

**Description**

Export multiLocalZscores slot from multiLocalZScore objects

**Usage**

```
mlzsMultiLocalZscores(x)
```

**Arguments**

|   |                            |
|---|----------------------------|
| x | A multiLocalZScore object. |
|---|----------------------------|

**Value**

multiLocalZscores slot of a multiLocalZScore object.

**See Also**

[multiLocalZScore](#)

---

|           |                                                        |
|-----------|--------------------------------------------------------|
| mlzsParam | <i>Export parameters from multiLocalZScore objects</i> |
|-----------|--------------------------------------------------------|

---

**Description**

Export parameters from multiLocalZScore objects

**Usage**

```
mlzsParam(x)
```

**Arguments**

|   |                            |
|---|----------------------------|
| x | A multiLocalZScore object. |
|---|----------------------------|

**Value**

parameters slot of a multiLocalZScore object.

**See Also**

[multiLocalZScore](#)

---

|             |                                                               |
|-------------|---------------------------------------------------------------|
| mlzsParam<- | <i>Set parameters slot value of a multiLocalZScore object</i> |
|-------------|---------------------------------------------------------------|

---

**Description**

Set parameters slot value of a multiLocalZScore object

**Usage**

```
mlzsParam(x) <- value
```

**Arguments**

|       |                                         |
|-------|-----------------------------------------|
| x     | A multiLocalZScore object.              |
| value | Value to assign to the parameters slot. |

**Value**

A multiLocalZScore object.

---

|              |                     |
|--------------|---------------------|
| mLZ_regA_ReG | <i>mLZ_regA_ReG</i> |
|--------------|---------------------|

---

**Description**

Alien Genome multiLocalZScore calculated for regA regionset from AlienRSList\_narrow using [regioneR::resampleGenome\(\)](#) function as permutation strategy.

**Usage**

```
data(cw_Alien)
```

**Format**

An objects of class [multiLocalZScore](#); see [makeLZMatrix\(\)](#).

---

|                 |                        |
|-----------------|------------------------|
| mLZ_regA_ReG_br | <i>mLZ_regA_ReG_br</i> |
|-----------------|------------------------|

---

**Description**

Alien Genome multiLocalZScore calculated for regA regionset from AlienRSList\_broad using [regioneR::resampleGenome\(\)](#) function as permutation strategy.

**Usage**

```
data(cw_Alien)
```

**Format**

An object of class [multiLocalZScore](#)

---

mLZ\_regD\_ReG

*mLZ\_regD\_ReG*


---

### Description

Alien Genome multiLocalZScore calculated for regD regionset from AlienRSList\_narrow using [regioneR::resampleGenome\(\)](#) function as permutation strategy.

### Usage

```
data(cw_Alien)
```

### Format

An object of class [multiLocalZScore](#)

---

multiLocalZscore

*multiLocalZscore*


---

### Description

Perform multiple permutation tests between a region set and each element in a list of region sets using shifted positions to calculate a local z-score.

### Usage

```
multiLocalZscore(A, Blist = NULL, sampling = FALSE, fraction = 0.15,
  min_sampling = 5000, ranFUN = "randomizeRegions", evFUN = "numOverlaps",
  ntimes = 100, adj_pv_method = "BH", genome = "hg19", universe = NULL,
  window = 1000, step = 100, ...)
```

### Arguments

|              |                                                                                                                                                                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A            | query region set for which to estimate local z-score values.                                                                                                                                                                                                              |
| Blist        | <a href="#">GRangesList</a> or list of region sets in any accepted formats by <a href="#">regioneR</a> package ( <a href="#">GRanges</a> , <a href="#">data.frame</a> etc.).                                                                                              |
| sampling     | logical, if TRUE the function will use only a sample of each element of Alist to perform the test as specified in fraction. (default = FALSE)                                                                                                                             |
| fraction     | logical, if sampling=TRUE, defines the fraction of the region sets used to perform the test. (default = 0.15)                                                                                                                                                             |
| min_sampling | numeric, minimum number of regions accepted after sampling is performed with the specified fraction. If the number of sampled regions is less than min_sampling, the number specified by min_sampling will be used as number of regions sampled instead. (default = 5000) |
| ranFUN       | character, the randomization strategy used for the test, see <a href="#">regioneR</a> . (default = "randomizeRegions")                                                                                                                                                    |
| evFUN        | character, the evaluation strategy used for the test, see <a href="#">regioneR</a> . (default = "numOverlaps")                                                                                                                                                            |

|                            |                                                                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ntimes</code>        | numeric, number of permutations used in the test. (default = 100)                                                                                   |
| <code>adj_pv_method</code> | character, the method used for the calculation of the adjusted p-value, to choose between the options of <code>p.adjust()</code> . (default = "BH") |
| <code>genome</code>        | character or <a href="#">GRanges</a> , genome used to compute the randomization. (default = "hg19")                                                 |
| <code>universe</code>      | region set to use as universe, used only when <code>regioneR::resampleRegions()</code> function is selected. (default = NULL)                       |
| <code>window</code>        | numeric, window (number of base pairs) in which the local z-score will be calculated. (default = 1000)                                              |
| <code>step</code>          | numeric, step (number of base pairs) by which will be estimated the local Z-score. (default = 100)                                                  |
| <code>...</code>           | further arguments to be passed to other methods.                                                                                                    |

## Details

This function performs multiple permutation tests between a single region set and each element in a list of region sets. For every pairwise combination, the evaluation step is repeated each time shifting the position of all the regions in the query region set by a fixed step inside a defined window (using `regioneR::localZScore()`). This produces a "local z-score" profile that can be indicative of the nature of the association between region sets. For example, an association can occur "centrally" if the z-score value drops sharply when sifting the region set. On the other hand, two region sets may have a peak of local z-score away from the central position if they happen to occur often at a regular distance, showing a "lateral" association.

## Value

A object of class `multiLocalZScore` containing three slots

- `@parameters`
- `@multiLocalZscores`
- `@matrix`

## See Also

[regioneR::localZScore\(\)](#)

## Examples

```
fakeGenome<- regioneR::toGRanges("chrF",1,1000)
regA <- regioneR::createRandomRegions(nregions = 10, length.mean = 10,
length.sd = 2,genome = fakeGenome)
regB <- regioneR::createRandomRegions(nregions = 10,length.mean = 10,
length.sd = 2,genome = fakeGenome)
regAs <-similarRegionSet(GR = regA,genome = fakeGenome, name = "A",
vectorPerc = seq(0.1,0.3,by =0.1))
regBs <-similarRegionSet(GR = regB,genome = fakeGenome, name = "B",
vectorPerc = seq(0.1,0.3,by =0.1))
ABList <- c(regAs,regBs)

mlz_ptAB <- multiLocalZscore(A = regA, Blist = ABList,
genome = fakeGenome, ntimes = 10)
summary(mlz_ptAB)
```

---

```
multiLocalZScore-class
```

*multiLocalZScore Class*

---

### Description

An S4 class for "multiLocalZScore" object.

### Slots

**parameters** List of parameters used to create the object

**multiLocalZscores** Results of multiple pairwise permutation tests on shifted region sets generated with `multiLocalZscore()`.

**matrix** List of numerical matrices containing local z-scores and correlation values generated with `makeLZMatrix()`.

### Examples

```
data("cw_Alien")

AlienRSList_narrow_small <- AlienRSList_narrow[c("regA","regB","regC")]

mlz_test <- multiLocalZscore(A = AlienRSList_narrow_small$regA, Blist = AlienRSList_narrow_small,
                             sampling = FALSE, genome = AlienGenome, per.chromosome = TRUE,
                             ranFUN = "resampleGenome", evFUN = "numOverlaps",
                             ntimes = 10, mc.cores = 2)

class(mlz_test)
```

---

```
multiPermTest
```

*multiPermTest*

---

### Description

multiPermTest

### Usage

```
multiPermTest (A, Blist, ranFUN, evFUN, uni, genome, rFUN, ntimes, adj_pv_method, ...)
```

### Arguments

|        |                                                                                                                           |
|--------|---------------------------------------------------------------------------------------------------------------------------|
| A      | Genomic Ranges or any accepted formats by <b>regioneR</b> package ( <code>GRanges</code> , <code>data.frame</code> etc.). |
| ranFUN | character, the randomization strategy used for the test, see <b>regioneR</b> . (default = "randomizeRegions")             |
| evFUN  | character, the evaluation strategy used for the test, see <b>regioneR</b> . (default = "numOverlaps")                     |

|               |                                                                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| uni           | region set to use as universe, used only when <code>regioneR::resampleRegions()</code> function is selected. (default = NULL)                       |
| genome        | character or <code>GRanges</code> , genome used to compute the randomization. (default = "hg19")                                                    |
| ntimes        | numeric, number of permutations used in the test. (default = 100)                                                                                   |
| adj_pv_method | character, the method used for the calculation of the adjusted p-value, to choose between the options of <code>p.adjust()</code> . (default = "BH") |
| ...           | further arguments to be passed to other methods.                                                                                                    |

### Details

Perform a multiple permutation test

### Value

a data frame object computed starting from results of `regioneR::permTest()` function

### See Also

`regioneR::permTest()`

---

|                     |                            |
|---------------------|----------------------------|
| plotCrosswiseDimRed | <i>plotCrosswiseDimRed</i> |
|---------------------|----------------------------|

---

### Description

Plot a visualization of a `genoMatriXeR` object (or matrix) using different dimensional reduction algorithms (PCA, tSNE and UMAP).

### Usage

```
plotCrosswiseDimRed(mPT, type = "PCA", GM_clust = NA, clust_met =
  "hclust", nc = 5, listRS = NULL, main = "", labSize = 2, emphasize = FALSE,
  labAll = FALSE, labMaxOverlap = 100, ellipse = TRUE, colPal = NULL,
  perplexity = 10, theta = 0.1, return_table = FALSE, return_plot = TRUE, ...)
```

### Arguments

|           |                                                                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mPT       | an object of class <code>genoMatriXeR</code> or a numerical matrix.                                                                                                      |
| type      | character, dimensional reduction algorithm to use ("PCA", "tSNE", "UMAP"). (default = "PCA")                                                                             |
| GM_clust  | numeric, vector of assigned clusters used to cluster the matrix. If NA, the matrix will be clustered using the method defined by <code>clust_met</code> . (default = NA) |
| clust_met | character, unsupervised cluster strategy used ( <code>hclust</code> , <code>kmeans</code> or <code>pam</code> ). (default = "hclust")                                    |
| nc        | numeric, number of clusters to define if using the default "kmeans" method. (default = 5)                                                                                |
| listRS    | list, a list of names of region sets of interest to be highlighted in the graph. (default = NULL)                                                                        |

|                                |                                                                                                                                         |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>main</code>              | character, title for the plot. (default = "")                                                                                           |
| <code>labSize</code>           | numeric, size for point labels in the plot. If 0, no labels will be plotted. (default = 2)                                              |
| <code>emphasize</code>         | logical, if TRUE, only the cluster in which the elements of <code>listRS</code> are present will be highlighted. (default = FALSE)      |
| <code>labAll</code>            | logical, if TRUE all data points are labelled, even if not in <code>listRS</code> when <code>emphasize</code> = TRUE. (default = FALSE) |
| <code>labMaxOverlap</code>     | numeric, max.overlaps for <code>geom_text_repel</code> . (default = 100)                                                                |
| <code>ellipse</code>           | logical, if TRUE ellipses will be drawn around the clusters. (default = FALSE)                                                          |
| <code>colPal</code>            | character, colors to use as palette for the plot. If NULL, default colors will be used. (default = NULL)                                |
| <code>perplexity, theta</code> | numeric, if type = "tSNE" values of perplexity and theta for the function <code>Rtsne()</code> . (default = 10)                         |
| <code>return_table</code>      | logical, if TRUE a table with the cluster assigned to each region is returned. (default = FALSE)                                        |
| <code>return_plot</code>       | logical, if TRUE a plot is returned. (default = TRUE)                                                                                   |
| <code>...</code>               | further arguments to be passed on to other methods                                                                                      |

## Details

This function generates a plot with a two-dimensional representation of the association data stored in a `genoMatriXeR` object by using either PCA, tSNE or UMAP transformations of the data. This function incorporates a clustering step and allows to highlight specific region sets of interest and the clusters they belong to. In addition to generating a plot, a table with the cluster assignments can be retrieved.

## Value

A ggplot object or a table with cluster assignments is returned.

## See Also

`crosswisePermTest()`

## Examples

```
data("cw_Alien")

cw_Alien_ReG <- makeCrosswiseMatrix(cw_Alien_ReG)

plotCrosswiseDimRed(cw_Alien_ReG, type = "PCA")

CDR_clust <- plotCrosswiseDimRed(cw_Alien_ReG, type = "UMAP", return_table = TRUE)

print(CDR_clust)
```

---

plotCrosswiseMatrix     *plotCrosswiseMatrix*


---

## Description

Plot matrix of associations/correlations stored in a [genoMatriXeR](#) object.

## Usage

```
plotCrosswiseMatrix(mPT, lineColor = NA, interpolate = FALSE, colMatrix =
"default", matrix_type = "association", cor = "row",
maxVal = NA, main = "", ord_mat = NULL)
```

## Arguments

|             |                                                                                                                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mPT         | an object of class <a href="#">genoMatriXeR</a> or a numerical matrix.                                                                                                                            |
| lineColor   | logical, color for the line grid delineating the tiles of the matrix plot. If NA, no line will be drawn. (default = NA)                                                                           |
| interpolate | logical, if TRUE the image will be interpolated using the function <a href="#">geom_raster()</a> . (default = FALSE)                                                                              |
| colMatrix   | character or vector of colors, if "default" will be used a default selection see..                                                                                                                |
| matrix_type | character, type of matrix to be plotted, either "association" or "correlation". (default = "association")                                                                                         |
| cor         | character, if matrix_type is "correlation", choose if the function <a href="#">cor()</a> will be executed on each "row" or "col" of the matrix. (default = "row")                                 |
| maxVal      | numeric, maximum absolute value displayed by the plot. If "max", the maximum values in the matrix are used. If NA, the 0.95 quantile of all absolute values is used. (default = NA)               |
| main        | character, title of the plot. (default = "")                                                                                                                                                      |
| ord_mat     | numeric, list with two numeric vectors that represent the ordering of rows and column of the matrix to be used in the plot. If NULL, the order of the matrix is preserved as is. (default = NULL) |

## Details

This functions creates a graphical representation of the matrix of associations stored in a [genoMatriXeR](#) object. The values plotted and clustering options can be controlled when creating the matrix with the function [makeCrosswiseMatrix](#).

## Value

Returns a ggplot object.

## See Also

[crosswisePermTest](#) [makeCrosswiseMatrix](#)

## Examples

```
data("cw_Alien")

cw_Alien_ReG <- makeCrosswiseMatrix( cw_Alien_ReG)

plotCrosswiseMatrix(cw_Alien_ReG, matrix_type = "association")

plotCrosswiseMatrix(cw_Alien_ReG, matrix_type = "correlation")
```

---

plotLocalZScoreMatrix *Plot Local Z-Score Matrix*

---

## Description

Plot Local Z-Score Matrix of associations/correlations stored in a [multiLocalZScore](#) object.

## Usage

```
plotLocalZScoreMatrix (mLZ, lineColor = NA, colMatrix = "default",
  matrix_type = "association", maxVal = "max", main = "", labSize = 6,
  revert = FALSE, highlight = NULL, highlight_size = 2.5, highlight_max = FALSE,
  smoothing = FALSE, ...)
```

## Arguments

|                |                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mLZ            | an object of class <a href="#">multiLocalZScore</a> or a matrix                                                                                                                     |
| lineColor      | logical, color for the line grid delineating the tiles of the matrix plot. If NA, no line will be drawn. (default = NA)                                                             |
| colMatrix      | character or vector of colors, if "default" will be used a default selection see..                                                                                                  |
| matrix_type    | character, type of matrix to be plotted, either "association" or "correlation". (default = "association")                                                                           |
| maxVal         | numeric, maximum absolute value displayed by the plot. If "max", the maximum values in the matrix are used. If NA, the 0.95 quantile of all absolute values is used. (default = NA) |
| main           | character, title of the plot. (default = "")                                                                                                                                        |
| labSize        | numeric, size for the plot labels. (default = 6)                                                                                                                                    |
| revert         | logical, if TRUE reverts the order of the plotted elements. (default = FALSE)                                                                                                       |
| highlight      | character, vector indicating the region set names to highlight by adding labels pointing to the 0 shift position (default = NULL)                                                   |
| highlight_size | numeric, size of the highlight labels. (default = 2.5)                                                                                                                              |
| highlight_max  | logical, if TRUE the highlight labels are placed at the maximum local z-score value instead of the 0 shift position. (default = FALSE)                                              |
| smoothing      | logical, if TRUE the <a href="#">stats::smooth.spline</a> function will be applied to the local z-score profile. (default = FALSE)                                                  |
| ...            | further arguments to be passed to other methods.                                                                                                                                    |

**Value**

Returns a ggplot object.

**See Also**

[multiLocalZscore](#) [makeLZMatrix](#) [multiLocalZScore](#)

**Examples**

```
data("cw_Alien")
```

---

plotSingleLZ

*plotSingleLZ*


---

**Description**

Plot the result of specific local Z-Score tests from a [multiLocalZScore](#) object in the form of line plot profiles.

**Usage**

```
plotSingleLZ(mLZ, RS, xlab = "", normZS = TRUE, ylim = NULL, main = NA,
  colPal = NULL, labValues = TRUE, labSize = 2.5, labMax = FALSE, smoothing = FALSE, ...)
```

**Arguments**

|           |                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mLZ       | an object of class <a href="#">multiLocalZScore</a> .                                                                                                                        |
| RS        | character, vector of region set names for which to plot the local Z-score results.                                                                                           |
| xlab      | character, label for the x axis. (default = NA)                                                                                                                              |
| normZS    | logical, indicates whether the normalized Z-score values should be plotted. If FALSE, the raw Z-score is used. (default = TRUE)                                              |
| ylim      | numeric, vector with two elements: minimum and maximum Y values of the plot. If NULL, the plot limits are set by default so all data points can be plotted. (default = NULL) |
| main      | character, title for the plot. If NA, the name of the query region set in the multi-LocalZScore object will be used. (default = NA)                                          |
| colPal    | character, colors to use as palette for the plot. If NULL, default colors will be used. (default = NULL)                                                                     |
| labValues | logical, if TRUE each local Z-score profile is labelled at position 0 with the name of the region set and its Z-score value at the central position. (default = TRUE)        |
| labSize   | numerical, size of the labels from labValues in the plot. (default = 2.5)                                                                                                    |
| labMax    | logical, if TRUE the labels are placed at the maximum value of each local Z-score profile instead of the center. (default = FALSE)                                           |
| smoothing | logical, if TRUE the <a href="#">smooth.spline()</a> function will be applied to the localZ-score profile. (default = FALSE)                                                 |
| ...       | further arguments to be passed to other methods.                                                                                                                             |

## Details

This function generates a line plot with the local Z-score profiles of selected region sets from a [multiLocalZScore](#) object. This type of plot complements the local Z-score matrix (generated by [plotLocalZScoreMatrix](#), since it allows to visualize in detail the local Z-score profile of just the region sets of interest.

This plot is well suited for a single or a few region sets, but will get busy if attempting to plot many different profiles. For the latter, the full matrix generated by [plotLocalZScoreMatrix](#) is usually a better visualization option.

## Value

Returns a ggplot object.

## See Also

[multiLocalZscore\(\)](#), [makeLZMatrix\(\)](#)

## Examples

```
data("cw_Alien")

plotSingleLZ(mLZ_regA_ReG, RS = c("regD", "regD_02", "regA", "regAB_04"),
labMax = TRUE, smoothing = TRUE)
```

---

plotSinglePT

*plotSinglePT*

---

## Description

Plot the result of a single pairwise permutation test from a [genoMatriXeR](#) object.

## Usage

```
plotSinglePT(mPT, RS1, RS2, xlab = NA, main = NA)
```

## Arguments

|          |                                                                                                                                        |
|----------|----------------------------------------------------------------------------------------------------------------------------------------|
| mPT      | an object of class <a href="#">genoMatriXeR</a> .                                                                                      |
| RS1, RS2 | character, names of region sets in a <a href="#">genoMatriXeR</a> object for which to represent the pairwise permutation test results. |
| xlab     | character, label for x axis. (default = NA)                                                                                            |
| main     | title for the plot, if NA the name of the <a href="#">genoMatriXeR</a> object is used (default = NA)                                   |

## Details

This function generates a plot representing the result of a single permutation test stored in a [genoMatriXeR](#) object. This includes a plot of the density distribution of the randomized evaluations and a vertical line showing the observed evaluation in the original region set. The values of the mean randomized evaluations and the value of the observed evaluation are shown, in addition to the calculated Z-score, normalized Z-score and adjusted p-value.

**Value**

Returns a ggplot object.

**See Also**

[crosswisePermTest](#) [makeCrosswiseMatrix](#)

**Examples**

```
data("cw_Alien")
plotSinglePT(cw_Alien_ReG, RS1 = "regA", RS2 = "regA_05")
plotSinglePT(cw_Alien_ReG, RS1 = "regA", RS2 = "regC")
```

---

print,genoMatriXeR-method

*Print method for genoMatriXeR class*

---

**Description**

Print method for genoMatriXeR class

**Usage**

```
## S4 method for signature 'genoMatriXeR'
print(x)
```

**Arguments**

x                      A genoMatriXeR object.

**Value**

A printed output for genoMatriXeR objects.

**See Also**

[genoMatriXeR](#)

---

|                      |                             |
|----------------------|-----------------------------|
| randomizeRegionsPerc | <i>randomizeRegionsPerc</i> |
|----------------------|-----------------------------|

---

## Description

Create a random region set similar to a reference region set.

## Usage

```
randomizeRegionsPerc(GR, genome = "hg19", frac = 0.2, ...)
```

## Arguments

|        |                                                                             |
|--------|-----------------------------------------------------------------------------|
| GR     | a <a href="#">GRanges</a> object with the input region set.                 |
| genome | genome of reference to generate the similar region sets. (default = "hg19") |
| frac   | fraction of the original region set to randomize. (default = 0.2)           |
| ...    | further arguments to be passed to other methods.                            |

## Details

This function takes an input region set and generates a region set where a fraction of the regions has been randomized.

## Value

a [GRanges](#) object

## See Also

[similarRegionSet\(\)](#)

## Examples

```
data("cw_Alien")

nreg <- 100

regA <-
  createRandomRegions(
    nregions = nreg,
    length.mean = 100,
    length.sd = 10,
    non.overlapping = TRUE,
    genome = AlienGenome
  )

regA_02 <- randomizeRegionsPerc(GR = regA, genome = AlienGenome, frac = 0.2)
```

---

|                  |                           |
|------------------|---------------------------|
| similarRegionSet | <i>similar RegionSets</i> |
|------------------|---------------------------|

---

## Description

Create a list of similar region sets to a reference region set.

## Usage

```
similarRegionSet(GR, name, genome, vectorPerc)
```

## Arguments

|            |                                                                                                                                                                                             |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GR         | a <a href="#">GRanges</a> object with the input region set.                                                                                                                                 |
| name       | character, name for the output region sets. The names will be generated by adding an underscore and the fraction of similarity after the name of each region set generated. (default = "A") |
| genome     | genome of reference to generate the similar region sets. (default = "hg19")                                                                                                                 |
| vectorPerc | numeric, vector of desired randomized fractions. (default = seq(.1,.9,.1))                                                                                                                  |

## Details

This function takes a region set as an input and a vector of desired randomized fractions. For each fraction value, a new region set will be generated where that fraction of the original regions in the input region set has been randomized. In effect, this creates region sets that are "similar" to a controlled degree to the original region set. This tool can be useful for validation purposes and its use in the demonstration of the usage of this package can be seen in the *RegionReloaded* vignette.

## Value

A list of [GRanges](#) objects.

## See Also

[GRanges](#)

## Examples

```
data("cw_Alien")

A<-createRandomRegions(nregions = 20, length.mean = 1000, length.sd = 100,
  genome = AlienGenome)

similAList <- similarRegionSet(GR = A, genome = AlienGenome,
  vectorPerc = seq(0.1,0.9,0.2), name = "test")

summary (similAList)

data("cw_Alien")

regA <- createRandomRegions(
```

```

    nregions = 100,
    length.mean = 10,
    length.sd = 5,
    genome = AlienGenome
  )

listRegA <- similarRegionSet(GR = regA, genome = AlienGenome)
summary(listRegA)

```

---

subList

*Sub List from region set list*


---

### Description

create a samples sublist starting from a list of Region Set

### Usage

```
subList(Alist, min_sampling, fraction)
```

### Arguments

|              |                                                                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alist        | GRangesList or list of Region Set of any accepted formats by <b>regioneR</b> package                                                                                 |
| min_sampling | numeric, minimum number of regions for each element of the list, if the number of regions is less than min_samplnig as subsempilng will be take the entire regionset |
| fraction     | numeric, percentage of regions selected for the subsampling                                                                                                          |

### Value

a sublist of GenomicRanges from an original GenomicRanges list

# Index

## \* datasets

AlienGenome, [3](#)  
AlienRSList\_broad, [3](#)  
AlienRSList\_narrow, [4](#)  
cw\_Alien, [8](#)  
cw\_Alien\_RaR, [9](#)  
cw\_Alien\_ReG, [9](#)  
cw\_Alien\_ReG\_no\_Square, [9](#)  
cw\_Alien\_ReR, [10](#)  
mLZ\_regA\_ReG, [21](#)

## \* function

multiPermTest, [24](#)

## \* internal

cleanCrosswiseMatrix, [5](#)  
crosswiseMatrix, [6](#)  
DFfromLZ, [10](#)  
gmxrMatrix, [14](#)  
gmxrMatrix<-, [15](#)  
gmxrMultiOverlaps, [15](#)  
gmxrParam, [16](#)  
mlzsMatrix, [19](#)  
mlzsMatrix<-, [19](#)  
mlzsMultiLocalZscores, [20](#)  
mlzsParam, [20](#)  
mlzsParam<-, [21](#)  
multiPermTest, [24](#)  
print, genoMatriXeR-method, [31](#)  
subList, [34](#)

AlienGenome, [3](#), [3](#), [4](#)

AlienRSList\_broad, [3](#)

AlienRSList\_narrow, [4](#)

chooseHclustMet, [4](#)

chooseHclustMet(), [16–18](#)

cleanCrosswiseMatrix, [5](#)

cor(), [27](#)

createRandomRegions(), [3](#), [4](#)

createUniverse, [6](#)

crosswiseMatrix, [6](#)

crosswisePermTest, [7](#), [14](#), [27](#), [31](#)

crosswisePermTest(), [5](#), [11](#), [17](#), [26](#)

cw\_Alien, [8](#)

cw\_Alien\_RaR, [9](#)

cw\_Alien\_ReG, [9](#)

cw\_Alien\_ReG\_no\_Square, [9](#)

cw\_Alien\_ReR, [10](#)

data.frame, [7](#), [22](#), [24](#)

DFfromLZ, [10](#)

dist(), [4](#), [16](#), [18](#)

genoMatriXeR, [6–17](#), [25–27](#), [30](#), [31](#)

genoMatriXeR-class, [11](#)

geom\_raster(), [27](#)

geom\_text\_repel, [26](#)

getHClust, [11](#)

getMatrix, [12](#)

getMultiEvaluation, [13](#)

getParameters, [13](#)

gMXR (genoMatriXeR-class), [11](#)

gmxrMatrix, [14](#)

gmxrMatrix<-, [15](#)

gmxrMultiOverlaps, [15](#)

gmxrParam, [16](#)

GRanges, [3](#), [4](#), [6](#), [7](#), [22–25](#), [32](#), [33](#)

GRangesList, [7](#), [22](#)

hclust, [4](#), [11](#), [17](#), [25](#)

hclust(), [4](#), [5](#), [16](#), [18](#)

kmeans, [25](#)

localZScore, [18](#)

makeCrosswiseMatrix, [12](#), [16](#), [27](#), [31](#)

makeCrosswiseMatrix(), [5](#), [8–11](#)

makeLZMatrix, [12](#), [18](#), [29](#)

makeLZMatrix(), [21](#), [24](#), [30](#)

mLZ\_regA\_ReG, [21](#)

mLZ\_regA\_ReG\_br, [21](#)

mLZ\_regD\_ReG, [22](#)

mLZS (multiLocalZScore-class), [24](#)

mlzsMatrix, [19](#)

mlzsMatrix<-, [19](#)

mlzsMultiLocalZscores, [20](#)

mlzsParam, [20](#)

mlzsParam<-, [21](#)

multiLocalZScore, [10–14](#), [18–23](#), [28–30](#)

multiLocalZscore, [22](#), [29](#)  
multiLocalZscore(), [13](#), [24](#), [30](#)  
multiLocalZScore-class, [24](#)  
multiPermTest, [24](#)  
multiPermTest(), [17](#)  
  
p.adjust(), [7](#), [23](#), [25](#)  
pam, [25](#)  
plotCrosswiseDimRed, [25](#)  
plotCrosswiseMatrix, [27](#)  
plotCrosswiseMatrix(), [17](#)  
plotLocalZScoreMatrix, [28](#), [30](#)  
plotSingleLZ, [29](#)  
plotSinglePT, [30](#)  
print, genoMatriXeR-method, [31](#)  
  
randomizeRegionsPerc, [32](#)  
regioneR::circularRandomizeRegions, [8](#)  
regioneR::joinRegions(), [6](#)  
regioneR::localZScore(), [23](#)  
regioneR::overlapPermTest(), [8](#)  
regioneR::permTest(), [7](#), [8](#), [25](#)  
regioneR::randomizeRegions, [8](#)  
regioneR::randomizeRegions(), [9](#)  
regioneR::resampleGenome, [8](#)  
regioneR::resampleGenome(), [9](#), [21](#), [22](#)  
regioneR::resampleRegions, [8](#)  
regioneR::resampleRegions(), [6](#), [7](#), [10](#), [23](#),  
[25](#)  
Rtsne(), [26](#)  
  
similarRegionSet, [33](#)  
similarRegionSet(), [3](#), [4](#), [32](#)  
smooth.spline(), [29](#)  
stats::smooth.spline, [28](#)  
subList, [34](#)  
  
t(), [16](#)