Package ‘FunciSNP’

February 13, 2012

Type Package

Title The Functional Integration of SNPs with Phenotype by Coincidence with Chromatin Biofeatures

Version 0.1.7

Date 2012-02-07

Author Simon G. Coetzee (scoetzee NEAR gmail POINT com) and Houtan Noushmehr (houtana NEAR gmail POINT com)

Maintainer Simon G. Coetzee <`scoetzee@gmail.com`>

biocViews Infrastructure, DataRepresentation, DataImport,SequenceMatching, Annotation

Depends
R (>= 2.14.0), Rsamtools (>= 1.6.1), rtracklayer(>= 1.14.1),GGtools (>= 4.0.0), methods, ChIPpeakAnno (>= 2.2.0),GenomicRanges, TxDb.Hsapiens.UCSC.hg19.knownGene,VariantAnnotation, plyr, org.Hs.eg.db, snp

Imports IRanges, AnnotationDbi

Suggests gplots (>= 2.10.1), ggplot2 (>= 0.8.9), matlab (>= 0.8.9)

Enhances parallel

Description FunciSNP integrates information from GWAS, 1000genomes and chromatin feature to identify functional SNP in coding or non-coding regions.

License GPL-3

LazyLoad yes

R topics documented:

`FunciSNP-package` *Functional Identification of SNPs with Phenotype by Coincidence with Chromatin Biofeatures*

Description

The package includes functions to identify and annotate putative functional SNPs using information derived from GWAS, 1000 genomes database, and sequences around peaks.

Details

Package:	FunciSNP
Type:	Package
Version:	0.1.7
Date:	2012-12-12
License:	GPL-3
LazyLoad:	yes

Author(s)

Simon Coetzee and Houtan Noushmehr

Maintainer: Simon G. Coetzee <scoetzee@gmail.com>

References

Coetzee SG et al. submitted for review. 2012

See Also

`FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
##
## Glioblastoma analysis using FunciSNP
##
```

```
## Full path to the example regions file for Glioblastoma
#  (collected from SNPedia)
glioma.snp <- file.path(system.file('extdata',
  package='FunciSNP'),
  dir(system.file('extdata',package='FunciSNP'),
  pattern='.snp$'));

## Full path to the example biological features BED files
#  derived from the ENCODE project for Glioblastoma U-87
#  cell lines.
glioma.bio <- system.file('extdata',package='FunciSNP');

## FunciSNP analysis, extracts correlated SNPs from the
#  1000 genomes db ("ncbi") and finds overlaps between
#  correlated SNP and biological features and then
#  calculates LD (Rsquare, Dprime, distance, p-value).
# Do not run. Can take more than 5 min depending on internet connection and number of CPUs.
#glioma <- FunciSNP(snp.regions.file=glioma.snp,
#  bio.features.loc = glioma.bio, bio.features.TSS=FALSE);

##
data(glioma);
class(glioma);
glioma;
summary(glioma);
```

CorrelatedSNPs-class *Class* "CorrelatedSNPs"

Description

Class for CorrelatedSNPs

Objects from the Class

Objects can be created by calls of the form `new("CorrelatedSNPs", ...)`.

Slots

`chromosome`: Object of class "integer" ~~

`position`: Object of class "integer" ~~

`snpid`: Object of class "character" ~~

`ref.allele`: Object of class "character" ~~

`alt.allele`: Object of class "character" ~~

`overlapping.features`: Object of class "GRanges" ~~

`genotype`: Object of class "CorrGeno" ~~

`ALL.R.squared`: Object of class "`matrix`" ~~

`AFR.R.squared`: Object of class "`matrix`" ~~

`AMR.R.squared`: Object of class "`matrix`" ~~

`ASN.R.squared`: Object of class "`matrix`" ~~

`EUR.R.squared`: Object of class "`matrix`" ~~

`ALL.D.prime`: Object of class "`matrix`" ~~

`AFR.D.prime`: Object of class "`matrix`" ~~

`AMR.D.prime`: Object of class "`matrix`" ~~

`ASN.D.prime`: Object of class "`matrix`" ~~

`EUR.D.prime`: Object of class "`matrix`" ~~

`ALL.p.value`: Object of class "`list`" ~~

`AFR.p.value`: Object of class "`list`" ~~

`AMR.p.value`: Object of class "`list`" ~~

`ASN.p.value`: Object of class "`list`" ~~

`EUR.p.value`: Object of class "`list`" ~~

Methods

AFR.D.prime<- `signature(x = "CorrelatedSNPs")`: ...

AFR.D.prime `signature(x = "CorrelatedSNPs")`: ...

AFR.p.value<- `signature(x = "CorrelatedSNPs")`: ...

AFR.p.value `signature(x = "CorrelatedSNPs")`: ...

AFR.R.squared<- `signature(x = "CorrelatedSNPs")`: ...

AFR.R.squared `signature(x = "CorrelatedSNPs")`: ...

ALL.D.prime<- `signature(x = "CorrelatedSNPs")`: ...

ALL.D.prime `signature(x = "CorrelatedSNPs")`: ...

ALL.p.value<- `signature(x = "CorrelatedSNPs")`: ...

ALL.p.value `signature(x = "CorrelatedSNPs")`: ...

ALL.R.squared<- `signature(x = "CorrelatedSNPs")`: ...

ALL.R.squared `signature(x = "CorrelatedSNPs")`: ...

alt.allele<- `signature(x = "CorrelatedSNPs")`: ...

alt.allele `signature(x = "CorrelatedSNPs")`: ...

AMR.D.prime<- `signature(x = "CorrelatedSNPs")`: ...

AMR.D.prime `signature(x = "CorrelatedSNPs")`: ...

AMR.p.value<- `signature(x = "CorrelatedSNPs")`: ...

AMR.p.value `signature(x = "CorrelatedSNPs")`: ...

AMR.R.squared<- `signature(x = "CorrelatedSNPs")`: ...

AMR.R.squared `signature(x = "CorrelatedSNPs")`: ...

- **ASN.D.prime<-** `signature(x = "CorrelatedSNPs")`: ...
- **ASN.D.prime** `signature(x = "CorrelatedSNPs")`: ...
- **ASN.p.value<-** `signature(x = "CorrelatedSNPs")`: ...
- **ASN.p.value** `signature(x = "CorrelatedSNPs")`: ...
- **ASN.R.squared<-** `signature(x = "CorrelatedSNPs")`: ...
- **ASN.R.squared** `signature(x = "CorrelatedSNPs")`: ...
- **chr<-** `signature(x = "CorrelatedSNPs")`: ...
- **chr** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.D.prime<-** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.D.prime** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.p.value<-** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.p.value** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.R.squared<-** `signature(x = "CorrelatedSNPs")`: ...
- **EUR.R.squared** `signature(x = "CorrelatedSNPs")`: ...
- **overlapping.features<-** `signature(x = "CorrelatedSNPs")`: ...
- **overlapping.features** `signature(x = "CorrelatedSNPs")`: ...
- **pop.genotype<-** `signature(x = "CorrelatedSNPs")`: ...
- **pop.genotype** `signature(x = "CorrelatedSNPs")`: ...
- **position<-** `signature(x = "CorrelatedSNPs")`: ...
- **position** `signature(x = "CorrelatedSNPs")`: ...
- **ref.allele<-** `signature(x = "CorrelatedSNPs")`: ...
- **ref.allele** `signature(x = "CorrelatedSNPs")`: ...
- **snpid<-** `signature(x = "CorrelatedSNPs")`: ...
- **snpid** `signature(x = "CorrelatedSNPs")`: ...

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
showClass("CorrelatedSNPs")
```

`CorrGeno-class` *Class* `"CorrGeno"`

Description

placeholder«««<

Objects from the Class

Objects can be created by calls of the form `new("CorrGeno", ...)`.

placeholder«««<

Slots

`SnpMatrix`: Object of class `"SnpMatrix"`

`populations`: Object of class `"list"` placeholder«««<

placeholder«««<

Methods

No methods defined with class "CorrGeno" in the signature. placeholder«««<

Note

placeholder«««<

Author(s)

placeholder«««<

References

placeholder«««<

See Also

placeholder«««<

Examples

```
showClass("CorrGeno")
```

`FunciSNPAnnotateSummary` *Genomic Annotation of Func-y-SNPs.*

Description

This will annotate all identified Func-y-SNP for it's distance to the nearest known TSS, whether it overlapps a known exon, intron, 5'UTR, 3'UTR, promoter, lincRNA or in gene desert (intergentic) regions.

Usage

```
FunciSNPAnnotateSummary(snp.list)
```

Arguments

`snp.list`	a FunciSNP object: `snp.list` represents the FunciSNP object output from FunciSNP. See `FuncySNP`.

Details

All known genomic features (exon, intron, 5'UTR, 3'UTR, promoter, lincRNA or in gene desert (intergentic)) are used to annotate the newly identified Func-y-SNP. Information described in this data.frame is used for all summary plots, table, and bed file generations.

Value

data.frame with rows for each correlated SNP.

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
data(glioma);
gl <- FunciSNPAnnotateSummary(glioma);
dim(gl)
head(gl)
names(gl)
```

FunciSNPbed *Creates a BED file to view Func-y-SNPs in your favorite genome browser*

Description

FunciSNPbed will output a BED file to a specified folder. The BED file is in standard UCSC Genome Browser format (http://genome.ucsc.edu/FAQ/FAQformat). Each tagSNP is colored black and each Func-y-SNP is colored red.

Usage

```
FunciSNPbed(dat, rsq, path = getwd(), filename = NULL)
```

Arguments

`dat` FunciSNP data.frame: `dat` is a data.frame object from FunciSNPAnnotateSummary. Need to run `FunciSNPAnnotateSummary` first.

`rsq` an interger (0-1): `rsq` is the Rsquared cutoff used to subset.

`path` a character: `path` is the path to the folder where to save the BED file. Default to getwd() or current working directory.

`filename` a character: `filename` is the name of the BED file. If NULL, filename is 'FunciSNP_results_rsq.RSQ value.bed'

Details

FunciSNPbed outputs a unique BED file which can be used to view in any genomic browser compatible with BED formats. To learn more about BED formats, see UCSC Genome Browser FAQ (http://genome.ucsc.edu/FAQ/FAQformat). Each tagSNP which is in LD to a corresponding Func-y-SNP overlapping at least one biofeature is colored black, while the Func-y-SNP is colored red. The initial position is provided by the first tagSNP and the first linked Func-y-SNP. We recommend using UCSC genome browser to view your BED files. This is useful so you can view all public and private tracks in relation to FunciSNP results.

Value

BED file is outputed as a tab-deliminated file in the specified 'path' folder. See example below.

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

FuncySNP, FunciSNPplot, FunciSNPAnnotateSummary, FunciSNPtable, FunciSNPbed

Examples

```
##
data(glioma);
glioma.anno <- FunciSNPAnnotateSummary(glioma);
FunciSNPbed(glioma.anno, rsq=0.9);
####
#Bed file "FunciSNP_results_rsq.0.9.bed" created successfully.
#(See folder: "/home/houtan/Downloads/")
#Total corSNP (RED):  15
#Total tagSNP (BLK):  1

#To view results, submit bed file as a
#  custom track in UCSC Genome Browser (genome.ucsc.edu),

#Now have fun with your new Func-y SNPs!!
####
```

FunciSNPidsFromSummary *coming soon.*

Description

placeholder«««<

Usage

```
FunciSNPidsFromSummary(dat, tagsnpid = NULL, num.features, rsq = 0)
```

Arguments

dat	placeholder«««<
tagsnpid	placeholder«««<
num.features	placeholder«««<
rsq	placeholder«««< placeholder«««<

Details

placeholder«««

Value

placeholder«««

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
## coming soon
```

`FunciSNPplot` *FunciSNPplot to visualize Func-y-SNP summary.*

Description

FunciSNPplot is a function developed to plot various types of plots to summarize and assist end-user in making informed discoveries of FunciSNP results. Plots can be stored in a folder for future reference.

Usage

```
FunciSNPplot(dat, rsq = 0, split = FALSE, splitbysnp = FALSE, tagSummary = FALSE, heatmap = FALSE, genomi
```

Arguments

- `dat` FunciSNP data.frame: `dat` is a data.frame object from FunciSNPAnnotateSummary. Need to run `FunciSNPAnnotateSummary` first.
- `rsq` an interger (0-1): `rsq` is the Rsquared cutoff used to subset.
- `split` logical: `split` will generate distribution plot of all Correlated SNPs by Rsquare values.
- `splitbysnp` logical: `splitbysnp` is similar to `split` but instead split the distribution by tagSNP.
- `tagSummary` logical: `tagSummary` Will output two plots per biofeature. The first one is a scatter plot showing the relationship between Rsquare and Distance to tagSNP for each Func-y-SNP. The second plot is a histogram distribution of number of correlated SNPs at each Rsquare value. Each set of plot is further divided by tagSNP. Best if used with `rsq` value.
- `heatmap` logical: `heatmap` correlation heatmap to visualize the number of correlated SNPs at each tagSNP overlapping each biological feature. Most informative if used with a `rsq` value.
- `genomicSum` logical: `genomicSum` Stacked bar chart summarizing all correlated SNPs for each of the identified genomie features (exon, intron, 5’UTR, 3’UTR, promoter, lincRNA or in gene desert (intergentic)). Most informative if used with a `rsq` value.
- `save` logical: `save` to save outputs to folder. Set at getwd(), in folder ’FunciSNP.VERSION/plots
- `pathplot` a character: `pathplot` is the path to the folder where to save the plots. Default to getwd() or current working directory.

Details

NA

Value

Plots are generated either in X11 or in specified folder.

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
data(glioma)
gl <- FunciSNPAnnotateSummary(glioma)
FunciSNPplot(gl)
FunciSNPplot(gl, rsq=0, genomicSum=TRUE, save=FALSE)
FunciSNPplot(gl, rsq=0.5, genomicSum=TRUE, save=FALSE)
# DO NOT RUN
#FunciSNPplot(gl, tagSummary=TRUE, rsq=0.5)
#
```

FunciSNPsummaryOverlaps

coming

Description

placeholder«««<

Usage

```
FunciSNPsummaryOverlaps(dat, rsq = 0)
```

Arguments

dat	placeholder«««<
rsq	placeholder«««< placeholder«««<

Details

placeholder«««<

Value

placeholder«««<

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
##coming soon.
```

FunciSNPtable — *Will output a summary report from FunciSNP at specified Rsquare cutoffs.*

Description

Using a specified Rsquare value (0-1) to subset the data, a table is generated which summarizes the total number of Func-y-SNPs, associated tagSNPs, and number of overlapping biofeatures.

Usage

```
FunciSNPtable(dat, rsq, geneSum = FALSE)
```

Arguments

`dat` FunciSNP data.frame: `dat` is a data.frame object from FunciSNPAnnotateSummary. Need to run `FunciSNPAnnotateSummary` first.

`rsq` an interger (0-1): `rsq` is the Rsquared cutoff used to subset.

`geneSum` logical: `geneSum` is set to FALSE. Setting to TRUE will output a list of Gene names which are nearest to the Func-y-SNP.

Details

Using a specified Rsquare value (0-1) to subset the data, a table is generated which summarizes the total number of Func-y-SNPs, associated tagSNPs, and number of overlapping biofeatures. This will provide user a first look at the total number of available Func-y-SNP at a particular Rsquare cutoff. If geneSum is set to TRUE, a list of gene names is reported instead.

Value

Standard output which summarizes FunciSNP results.

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

FuncySNP, FunciSNPplot, FunciSNPAnnotateSummary, FunciSNPtable, FunciSNPbed

Examples

```
data(glioma);
gl <- FunciSNPAnnotateSummary(glioma);
FunciSNPtable(gl, rsq=0.5);
FunciSNPtable(gl, rsq=0.5, geneSum=TRUE);
```

FuncySNP *Functional Identification of SNPs with Phenotype by Coincidence with Chromatin Biofeatures*

Description

Given a set of known tag-SNPs associated with a particular phenotype (e.g. disease, trait), and a set of available biological features (e.g. peaks derived from ChIP-seq experiments for phenotype), returns correlated SNPs (from the 1000 genomes db) which are in linkage disequilibrium (LD) to a known disease associated tag-SNP and overlaps chromatin biological features. These identified correlated SNPs are characterized as putative functional SNPs for a particular trait.

Usage

```
FuncySNP(snp.regions.file, bio.features.loc = NULL,
                    bio.features.TSS = TRUE,
                    par.threads=detectCores()/2,
                    verbose = par.threads < 2, method.p = "BH",
                    reduce.by.features = TRUE, search.window = 200000)
```

Arguments

`snp.regions.file`

path: Location of the regions file: Regions file is tab-deliminated and contains three elements per row. First element defines the genomic location of the tagSNP, 'chr:position' (e.g. 5:5420030). Second element contains the tagSNP name, 'rsID' (e.g. rs6010620). Third element defines the 'POPULATION' (ASN, EUR, AFR, ALL) where the tagSNP was identified (e.g. ASN, EUR, AFR, ALL).

SNP Region file is imported and each row element (tagSNP element) is parsed for tagSNP name (rsXXXX), population (ASN, EUR, AFR, or ALL), and genomic location. Genomic location is used to define the window size (see 'search.window' argument). See example file here: file.path(system.file('data',package='FunciSNP'), dir(system.file('data',package='FunciSNP'), pattern='.snp$'));

`bio.features.loc`
: path: Location of the biological features folder: Each biological feature for a particular genomic phenotype should be separated as individual BED files (tab deliminated file with chr, start and end). See UCSC for more information about BED formats `http://genome.ucsc.edu/FAQ/FAQformat.html#format1`. See example below. Default set to NULL.

`bio.features.TSS`
: logical: To include promoter regions as an additional biofeature in the analysis. Promoters defined as -1000 to +100 bp of a known TSS. File extracted on Feb. 9, 2012 from UCSC genome table browser. Default set to TRUE.

`par.threads`
: an integer: Number of CPU cores to use for FunciSNP analysis. Default set at detectCores()/2. If par.threads > 1, then by default "verbose" = FALSE.

`verbose`
: logical: If set to TRUE, then regardless of par.threads value, all verbose message will output to terminal. If set to FALSE, no verbose message will output to terminal, except for warnings(). Default setting depends on number of 'par.threads' value.

`method.p`
: method: p-value correction (or adjustment) method (see ?p.adjust). Default set at "BH" (Benjamini & Hochberg (1995)).

`reduce.by.features`
: logical: If set to TRUE, then only correlated SNPs overlapping biological features will be filtered and used to calculate Rsquared, Dprime, distance and p-value. In addition, only these correlated SNPs will be used to generate plots and summary analysis. If set to FALSE, all correlated SNPs regardless of overlap with biological features will have an associated Rsquare, Dprime, distance and p-value associated with the tag-SNP as defined by the 'search.window'.

`search.window`
: an integer: genomic window size used to extract all available correlated SNPs from the 1000 genomes db. The window size is centered around the tagSNP position as defined in the regions.file.

Details

This is the main funtion of FunciSNP. It will identify correlated SNPs which are in linkage disequilibrium (LD) to a known disease associated tagSNP. It will also determine if the correlated SNP in LD to the tagSNP overlaps a genomic biological feature. Correlated SNPs are directly imported from the current public release of the 1000 genomes database. 1000 genomes ftp servers available for the 1000 genomes public data: 1) National Center for Biotechnology Information (NCBI) `ftp://ftp-trace.ncbi.nih.gov/1000genomes/`; 2) European Bioinformatics Institute (EBI) `ftp://ftp.1000genomes.ebi.ac.uk/vol1/`.

Correlated SNPs in LD to a tagSNP and overlapping genomic biological features are known as putative functional SNPs (also defined as 'Func-y-SNP' elsewhere in the package.).

Value

`TSList`
: FunciSNP object.

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

FunciSNPplot, FunciSNPAnnotateSummary, FunciSNPtable, FunciSNPbed

Examples

```
##
## Glioblastoma analysis using FunciSNP
##
## Full path to the example regions file for Glioblastoma
#  (collected from SNPedia)
glioma.snp <- file.path(system.file('extdata',
  package='FunciSNP'),
  dir(system.file('extdata',package='FunciSNP'),
  pattern='.snp$'));

## Full path to the example biological features BED files
#  derived from the ENCODE project for Glioblastoma U-87
#  cell lines.
glioma.bio <- system.file('extdata',package='FunciSNP');

## FunciSNP analysis, extracts correlated SNPs from the
#  1000 genomes db ("ncbi") and finds overlaps between
#  correlated SNP and biological features and then
#  calculates LD (Rsquare, Dprime, distance, p-value).
# Do not run. Can take more than 5 min depending on internet connection and number of CPUs.
#glioma <- FuncySNP(snp.regions.file=glioma.snp,
#  bio.features.loc = glioma.bio, bio.features.TSS=FALSE);

##
data(glioma);
class(glioma);
glioma;
summary(glioma);
```

`TagSNP-class` *Class* `"TagSNP"`

Description

hello

Objects from the Class

Objects can be created by calls of the form `new("TagSNP", ...)`.

Slots

`chromosome`**:** Object of class `"integer"` ~~

`position`**:** Object of class `"integer"` ~~

`snpid`**:** Object of class `"character"` ~~

`population`**:** Object of class `"character"` ~~

`ref.allele`**:** Object of class `"character"` ~~

`alt.allele`**:** Object of class `"character"` ~~

`overlapping.features`**:** Object of class `"GRanges"` ~~

`genotype`**:** Object of class `"SnpMatrix"` ~~

`R.squared.corrsnps`**:** Object of class `"dgCMatrix"` ~~

`D.prime.corrsnps`**:** Object of class `"dgCMatrix"` ~~

`correlated.snps`**:** Object of class `"CorrelatedSNPs"` ~~

Methods

AFR.overlapping.snps.geno `signature(object = "TagSNP")`: ...

ALL.overlapping.snps.geno `signature(object = "TagSNP")`: ...

alt.allele<- `signature(x = "TagSNP")`: ...

alt.allele `signature(x = "TagSNP")`: ...

AMR.overlapping.snps.geno `signature(object = "TagSNP")`: ...

ASN.overlapping.snps.geno `signature(object = "TagSNP")`: ...

chr<- `signature(x = "TagSNP")`: ...

chr `signature(x = "TagSNP")`: ...

correlated.snps<- `signature(x = "TagSNP")`: ...

correlated.snps `signature(x = "TagSNP")`: ...

D.prime.corrsnps<- `signature(x = "TagSNP")`: ...

D.prime.corrsnps `signature(x = "TagSNP")`: ...

EUR.overlapping.snps.geno `signature(object = "TagSNP")`: ...

genotype<- `signature(x = "TagSNP")`: ...

genotype `signature(x = "TagSNP")`: ...

overlapping.features<- `signature(x = "TagSNP")`: ...

overlapping.features `signature(x = "TagSNP")`: ...

population<- `signature(x = "TagSNP")`: ...

population `signature(x = "TagSNP")`: ...

position<- `signature(x = "TagSNP")`: ...

position `signature(x = "TagSNP")`: ...

ref.allele<- `signature(x = "TagSNP")`: ...

ref.allele `signature(x = "TagSNP")`: ...

R.squared.corrsnps<- `signature(x = "TagSNP")`: ...

R.squared.corrsnps `signature(x = "TagSNP")`: ...

show `signature(object = "TagSNP")`: ...

snpid<- `signature(x = "TagSNP")`: ...

snpid `signature(x = "TagSNP")`: ...

Note

NA

Author(s)

Simon Coetzee, Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
showClass("TagSNP")
```

TSList-class *Class* "TSList"

Description

ffff

Objects from the Class

Objects can be created by calls of the form `new("TSList", ...)`.

Slots

`snp.data`**:** Object of class "list" ~~

`summary.data`**:** Object of class "data.frame" ~~

`elementType`**:** Object of class "character" ~~

`elementMetadata`**:** Object of class "DataTableORNULL" ~~

`metadata`**:** Object of class "list" ~~

Methods

show `signature(object = "TSList")`: ...

summary `signature(object = "TSList")`: ...

Note

NA

Author(s)

Simon G. Coetzee and Houtan Noushmehr

References

Coetzee SG et al. submitted for review. 2012

See Also

`FuncySNP`, `FunciSNPplot`, `FunciSNPAnnotateSummary`, `FunciSNPtable`, `FunciSNPbed`

Examples

```
showClass("TSList")
```

Index