Package 'GGtools'

September 24, 2012

Title software and data for analyses in genetics of gene expression

Version 4.4.0

Author VJ Carey <stvjc@channing.harvard.edu>

Description software and data for analyses in genetics of gene expression

Suggests GGdata, illuminaHumanv1.db

Depends R (>= 2.14), stats4, GGBase (>= 3.16.1), IRanges,GenomicRanges, Rsamtools

Imports
methods, utils, stats, BiocGenerics, snpStats, ff,AnnotationDbi, Biobase, bit, VariantAnnotation

Enhances MatrixEQTL

Maintainer VJ Carey <stvjc@channing.harvard.edu>

License Artistic-2.0

LazyLoad yes

Collate AllClasses.R AllGenerics.R eqtlTests.R managers.R topFeats.R
gwSnpTests.R snpsCisToGenes.R relocate.R topSnps.R transutils.R
vcfutils.R eqtlEstimates.R alleq.R meta.R eqME.R

R topics documented:

GGtools-package *software and data for analyses in genetics of gene expression*

Description

software and data for analyses in genetics of gene expression

Details

Package:	GGtools
Version:	4.2.26
Suggests:	GGdata, illuminaHumanv1.db
Depends:	R (>= 2.14), GGBase (>= 3.16.1)
Imports:	methods, snpStats, ff, IRanges, GenomicRanges, AnnotationDbi, Biobase, Rsamtools, bit, VariantAnnotatio
License:	Artistic-2.0
LazyLoad:	yes
Packaged:	2012-01-18 03:39:51 UTC; stvjc
Collate:	AllClasses.R AllGenerics.R eqtlTests.R managers.R topFeats.R gwSnpTests.R snpsCisToGenes.R relocate.R
Built:	R 2.15.0; ; 2012-02-06 17:22:52 UTC; unix

Index:

best.cis.eQTLs	collect genewise best scoring eQTL
eqtlTests	compute association statistics between all probes and SNP in an smlSet instance
eqtlTestsManager-class	Class '"eqtlTestsManager"'
ex	ExpressionSet instance for illustrating integrative smlSet container
getCisMap	create, using Bioconductor annotation resources, a structure that enumerates SNP in the vicinity of ('cis' to) genes
gwSnpTests	execute a series of tests for association between genotype and expression
strMultPop	serialization of a table from Stranger's multipopulation eQTL report

The package depends on GGBase, which includes additional infrastructure for integrative data structures and data filtering.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Maintainer: VJ Carey <stvjc@channing.harvard.edu>

See Also

getSS for acquiring containers for integrative data on genetics of expression.

Examples

```
## Not run:
 # acquire chromosome 20 genotypes and all expression data for
 # 90 CEU samples as published at Wellcome Trust GENEVAR and
 # HapMap phase II
 c20 = getSS("GGtools", "20")
 # perform a focused eQTL search
 t1 = gwSnpTests(genesym("CPNE1")~male, c20)
 # get best hits
 topSnps(t1)

## End(Not run)
```

`best.cis.eQTLs` *collect genewise best scoring eQTL*

Description

collect genewise best scoring eQTL

Usage

```
best.cis.eQTLs(smpack = "GGdata", rhs = ~1,
  folderstem = "cisScratch", radius = 50000,
  shortfac = 100,
  chrnames = as.character(1:22),
  smchrpref = "", gchrpref = "", schrpref = "ch",
  geneApply = lapply, geneannopk = "illuminaHumanv1.db",
  snpannopk = "SNPlocs.Hsapiens.dbSNP.20100427",
  smFilter = function(x) nsFilter(MAFfilter(x, lower = 0.05), var.cutoff = 0.97), nperm = 2,
  useME=FALSE)

All.cis.eQTLs(maxfdr = 0.05, inbestcis = NULL, smpack = "GGdata",
    rhs = ~1, folderstem = "cisScratch", radius = 50000,
    shortfac = 100,
    chrnames = as.character(1:22),
    smchrpref = "", gchrpref = "", schrpref = "ch",
    geneApply = lapply, geneannopk = "illuminaHumanv1.db",
    snpannopk = "SNPlocs.Hsapiens.dbSNP.20100427",
    smFilter4cis = function(x) nsFilter(MAFfilter(clipPCs(x,
        1:10), lower = 0.05), var.cutoff = 0.85),
    smFilter4all = function(x) MAFfilter(clipPCs(x,
        1:10), lower = 0.05),
    nperm = 2)

meta.best.cis.eQTLs(smpackvec = c("GGdata", "hmyriB36"), rhslist = list(~1,
    ~1), folderstem = "cisScratch", radius = 50000, shortfac = 100,
    chrnames = as.character(1:22), smchrpref = "", gchrpref = "",
    schrpref = "ch", geneApply = lapply, geneannopk = "illuminaHumanv1.db",
    snpannopk = "SNPlocs.Hsapiens.dbSNP.20100427", smFilter = function(x) nsFilter(MAFfilter(x,
        lower = 0.05), var.cutoff = 0.97), nperm = 2)
```

```
meta.All.cis.eQTLs(maxfdr = 0.05, inbestcis = NULL, smpackvec = c("GGdata", "hmyriB36"),
  rhslist = list(~1, ~1), folderstem = "cisScratch",
  radius = 50000, shortfac=100, chrnames = as.character(1:22), smchrpref = "",
  gchrpref = "", schrpref = "ch", geneApply = lapply,
  geneannopk = "illuminaHumanv1.db",
  snpannopk = "SNPlocs.Hsapiens.dbSNP.20100427",
  smFilter4cis = function(x) nsFilter(MAFfilter(clipPCs(x, 1:10),
  lower = 0.05), var.cutoff = 0.85),
  smFilter4all = function(x) MAFfilter(clipPCs(x, 1:10),
  lower = 0.05),
  nperm = 2)


chromsUsed(x)

fdr(x)

fullreport(x, type)

getAll(x)

getBest(x)

getCall(x)
```

Arguments

smpack	character string naming a package to which getSS can be applied to extract smlSet-class instances
smpackvec	vector of character strings naming packages that can be used as smpack values in a series of best.cis.eQTLs calls, one per population for meta-analysis
rhs	R model formula, with no dependent variable, that will be used with snp.rhs.tests to adjust GWAS tests for each expression probe
rhslist	a list of model formulae to be used as rhs in a series of best.cis.eQTLs calls, one per population for meta-analysis
folderstem	prefix of the folder name to be used to hold ff archives of test results
radius	coding extent of each gene will be extended in both directions by radius bases, and only SNP within these limits are used for selecting best hits for the gene
shortfac	a numeric that will scale up the chi-squared statistic before it is converted to short integer for storage in ff array
chrnames	character vector of chromosome identifiers, to be manipulated for certain query resolutions by the following parameters
smchrpref	prefix to convert chrnames into appropriate tokens for indexing smlSet elements as collected from the package named by parameter smpack
gchrpref	prefix to convert chrnames into appropriate tokens for obtaining gene metadata; in future this may need to be a string transformation function
schrpref	prefix to convert chrnames into appropriate tokens for use with getSNPlocs for the SNP location information package identified in snpannopack parameter below

geneApply	an lapply like function, defaults to lapply
geneannopk	character string, name of annotation package that annotates probe identifiers
snpannopk	character string, name of SNPlocs.Hsapiens.dbSNP.* package for obtaining
smFilter	function accepting and returning an smlSet-class instance
nperm	number of permutations to be used for plug-in FDR computation
useME	logical; if TRUE, use the rudimentary interface to the MatrixEQTL package from A. Shabalin on CRAN
maxfdr	Used in All.cis.eQTLs. The process of identifying "best" cis eQTL per probe leads to a probe-specific FDR. In All.cis.eQTLs we enumerate all probes and all SNP with FDR at most maxfdr, not just the best scoring SNP per probe.
inbestcis	Used in All.cis.eQTLs. An instance of mcwBestCis that can be used to speed up the extraction of All.cis eQTL.
smFilter4cis	Used in All.cis.eQTLs. A function accepting and returning an smlSet instance. When inbestcis parameter is NULL, this filter will be used for identifying the best SNP per probe.
smFilter4all	Used in All.cis.eQTLs. A function accepting and returning an smlSet instance. This filter will be used for identifying the best SNP per probe. This filter should not affect the number of probes.
x	instance of mcwBestCis
type	character, either 'data.frame' or 'GRanges'

Details

geneApply can be set to parallel::mclapply, for example, in a multicore context.

mcwBestCis stands for 'multi-chromosome-wide best cis' eQTL report container.

It is possible that the filtering processes should be broken into genotype filtering and expression probe filtering.

fdr(x) will return a numeric vector of plug-in FDR estimates corresponding to probe:association tests as ordered in the fullreport of a *Cis container. More metadata should be attached to the output of this function.

Value

an instance of mcwBestCis

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
getClass("mcwBestCis")
## Not run:
best.cis.eQTLs(chrnames="20")

## End(Not run)
```

eqtlTests *compute association statistics between all probes and SNP in an smlSet instance*

Description

compute association statistics (or point estimates and standard errors) between all probes and SNP in an smlSet instance, using out-of-memory storage

Usage

```
eqtlTests(smlSet, rhs = ~1 - 1, runname = "foo",
targdir = "foo", geneApply = lapply,
shortfac = 100,
checkValid = TRUE, useUncertain = TRUE,
glmfamily = "gaussian")

eqtlEstimates(smlSet, rhs = ~1 - 1, runname = "foo",
targdir = "fooe", geneApply = lapply,
shortfac = 10000,
checkValid = TRUE, useUncertain = TRUE,
glmfamily = "gaussian")
```

Arguments

smlSet	instance of smlSet
rhs	fragment of a standard formula, minus a dependent variable (i.e., starts with tilde); bindings will be sought in pData(smlSet)
runname	string used to identify output ff files
targdir	string naming the folder where ff outputs will reside
geneApply	analog to lapply to drive iteration over probes
shortfac	ff contents will be multiplied by this quantity and stored as short integers
checkValid	logical, will apply validObject to smlSet if TRUE
useUncertain	logical, passed as uncertain parameter to snp.rhs.tests to specify whether uncertain genotypes will be used (as 'dosage' in GLM fitting)
glmfamily	family specification for snp.rhs.tests

Details

The purpose of the eqtlTests function is to allow very substantial eQTL search processes to occur with R. For several million SNP and tens of thousands of probes, the storage of test results requires attention to parsimony. The storage occurs out of memory, using the ff package, and employs short integers to represent chi squared statistics. These are scaled up prior to storage, and will be scaled down prior to use.

eqtlEstimates will use compact storage for both the point estimates and standard errors of association estimated under an additive genetic model

Value

returns an instance of `eqtlTestsManager`

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
hm2ceuSMS = getSS("GGtools", c("20"), renameChrs=c("chr20"))
library(illuminaHumanv1.db)
cptag = get("CPNE1", revmap(illuminaHumanv1SYMBOL))
indc = which(featureNames(hm2ceuSMS) == cptag[1])
#
# get a set of additional genes on chr20
all20 = get("20", revmap(illuminaHumanv1CHR))
g20 = unique(c(all20[1:10], cptag))
#
hm = hm2ceuSMS[probeId(g20),]  # reduce problem
td = tempdir()
curd = getwd()
setwd(td)
time.lapply = unix.time(e1 <- eqtlTests( hm, ~male ))
time.lapply
e1
# best chisq(1) for CPNE1
topFeats(probeId(cptag), e1)
setwd(curd)
```

---

eqtlTestsManager-class

Class "eqtlTestsManager"

Description

manage out-of-memory elements of an eQTL search

Objects from the Class

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

Slots

`fffile`: Object of class "ff_matrix" chisquared statistics stored as short ints in ff out of memory file

`call`: Object of class "call" audit of creation call

`sess`: Object of class "ANY" session info structure at time of creation

`exdate`: Object of class "ANY" date at time of creation

`shortfac`: Object of class "numeric" number by which chisq stats are multiplied to allow recovery of precision

`geneanno`: Object of class `"character"` string naming annotation package relevant for probe identifier translation

`df`: Object of class `"numeric"` degrees of freedom of chisq stats

`summaryList`: Object of class `"list"` list of genotype statistical summaries

Methods

[`signature(x = "eqtlTestsManager", i = "ANY", j = "ANY", drop = "ANY")`: extract chisq statistics properly rescaled from short int to double

show `signature(object = "eqtlTestsManager")`: concise report

topFeats `signature(feat = "probeId", mgr = "eqtlTestsManager")`: extract highest scores for SNP associated with given probeId

topFeats `signature(feat = "rsid", mgr = "eqtlTestsManager")`: extract highest scores for probes associated with given SNP

Note

instances are created by `eqtlTests`

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
showClass("eqtlTestsManager")
```

`ex` *ExpressionSet instance for illustrating integrative smlSet container*

Description

ExpressionSet instance for illustrating integrative smlSet container

Usage

```
data(eset)
```

Format

The format is: Formal class 'ExpressionSet' [package "Biobase"] with 7 slots ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots
.. .. ..@ name : chr ""
.. .. ..@ lab : chr ""
.. .. ..@ contact : chr ""
.. .. ..@ title : chr ""
.. .. ..@ abstract : chr ""
.. .. ..@ url : chr ""
.. .. ..@ pubMedIds : chr ""
.. .. ..@ samples : list()
.. .. ..@ hybridizations : list()

.. .. ..@ normControls : list()
.. .. ..@ preprocessing : list()
.. .. ..@ other : list()
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. ..@ .Data:List of 2
.. ..$: int [1:3] 1 0 0
.. ..$: int [1:3] 1 1 0
..@ assayData :<environment: 0x10bf12948>
..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata :'data.frame': 7 obs. of 1 variable:
.. ..$ labelDescription: chr [1:7] "hapmap family id" "hapmap person id" "id of mother of this person" "id of father of this person" ...
.. .. ..@ data :'data.frame': 90 obs. of 7 variables:
.. ..$ famid : int [1:90] 1341 1341 1341 1340 1340 1340 1340 1340 1341 1341 ...
.. ..$ persid : int [1:90] 14 2 13 9 10 2 11 1 11 1 ...
.. ..$ mothid : int [1:90] 0 14 0 0 0 12 0 10 0 12 ...
.. ..$ fathid : int [1:90] 0 13 0 0 0 11 0 9 0 11 ...
.. ..$ sampid : Factor w/ 90 levels "NA06985","NA06991",..: 1 2 3 4 5 6 7 8 9 10 ...
.. ..$ isFounder: logi [1:90] TRUE FALSE TRUE TRUE TRUE FALSE ...
.. ..$ male : logi [1:90] FALSE FALSE TRUE TRUE FALSE FALSE ...
.. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns"
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. ..@ .Data:List of 1
.. ..$: int [1:3] 1 1 0
..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable:
.. ..$ labelDescription: chr(0)
.. .. ..@ data :'data.frame': 47293 obs. of 0 variables
.. .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns"
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. ..@ .Data:List of 1
.. ..$: int [1:3] 1 1 0
..@ annotation : chr "illuminaHumanv1.db"
..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable:
.. ..$ labelDescription: chr(0)
.. .. ..@ data :'data.frame': 90 obs. of 0 variables
.. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns"
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. ..@ .Data:List of 1
.. ..$: int [1:3] 1 1 0
..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. ..@ .Data:List of 4
.. ..$: int [1:3] 2 14 0
.. ..$: int [1:3] 2 13 7
.. ..$: int [1:3] 1 3 0
.. ..$: int [1:3] 1 0 0

Details

Expression data harvested in 2007 from GENEVAR

ftp://ftp.sanger.ac.uk/pub/genevar/CEU_parents_norm_march2007.zip

Examples

```
data(eset) # yields ExpressionSet instance called ex
```

`getCisMap` *create, using Bioconductor annotation resources, a structure that enumerates SNP in the vicinity of ('cis' to) genes*

Description

create a structure that enumerates SNP in the vicinity of ('cis' to) genes

Usage

```
getCisMap(radius = 50000, gchr = "20",
  schr = "ch20", geneannopk = "illuminaHumanv1.db",
  snpannopk = "SNPlocs.Hsapiens.dbSNP.20100427",
  as.GRangesList = FALSE)
```

Arguments

`radius`	How far, in bases, up or down stream from the asserted coding region limits to include SNP
`gchr`	the token to be used to acquire locations for probes on a specified chromosome, using `revmap([dbpk]CHR)`
`schr`	the token to be used to acquire locations for SNP on a specified chromosome, using `getSNPlocs`
`geneannopk`	character string naming a Bioconductor .db expression chip annotation package
`snpannopk`	character string naming a Bioconductor SNPlocs.* SNP metadata package
`as.GRangesList`	logical telling whether a GRangesList should be returned

Details

This is a utility that the developer would rather not export. The complexity of harmonizing queries among probe and SNP annotation resources leads to a somewhat fragile product. Users who have their own gene ranges and SNP locations can examine the namelist component of the output of the default call to see what is expected for the *.cis.eQTLs function. For the set of chromosomes to be analyzed, there will be a list of chromosome specific namelist-like lists.

Value

Instance of `cisMap` class, which will retain SNP location, gene range, and radius information for auditing.

Examples

```
 ## Not run:
  getCisMap()

## End(Not run)
```

gwSnpTests *execute a series of tests for association between genotype and expression*

Description

execute a series of tests for association between genotype and expression

Usage

```
gwSnpTests(sym, sms, ...)
topSnps(x, n=10)
```

Arguments

sym	instance of probeId or genesym
sms	instance of smlSet-class
x	instance of gwSnpScreenResult
n	integer, number of test results to be reported, sorted decreasing from the most significant
...	not used

Details

The plot method for gwSnpScreenResult instances takes a second argument, the name of a Bioconductor SNPlocs.* package.

Value

an instance of the gwSnpScreenResult class, to be examined by topSnps

Note

The most basic application yields one d.f. chi-squared statistics based on score tests.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
s20 = getSS("GGtools", "20")
t1 = gwSnpTests(genesym("CPNE1")~male, s20)
topSnps(t1)
## Not run:
plot(t1, "SNPlocs.Hsapiens.dbSNP.20100427")

## End(Not run)
```

strMultPop *serialization of a table from Stranger's multipopulation eQTL report*

Description

serialization of a table from Stranger's multipopulation eQTL report

Usage

```
data(strMultPop)
```

Format

A data frame with 39649 observations on the following 12 variables.

`rsid` a factor with levels rs...

`genesym` a factor with levels `37865` `39692` `ABC1` `ABCD2` `ABHD4` `ACAS2` ...

`illv1pid` a factor with levels `GI_10047105-S` `GI_10092611-A` `GI_10190705-S` `GI_10567821-S` `GI_10835118-S` `GI_10835186-S` ...

`snpChr` a numeric vector

`snpCoordB35` a numeric vector

`probeMidCoorB35` a numeric vector

`snp2probe` a numeric vector

`minuslog10p` a numeric vector

`adjR2` a numeric vector

`assocGrad` a numeric vector

`permThresh` a numeric vector

`popSet` a factor with levels `CEU-CHB-JPT` `CEU-CHB-JPT-YRI` `CHB-JPT`

Details

imported from the PDF(!) distributed by Stranger et al as supplement to PMID 17873874

Source

PMID 17873874 supplement

References

PMID 17873874 supplement

Examples

```
data(strMultPop)
strMultPop[1:2,]
```

transManager-class *Class* "transManager"

Description

simple container for manager of transScores output

Objects from the Class

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

Slots

base: Object of class "list" includes ff references for scores and indices of genes corresponding to scores, and other metadata about the run

Methods

show signature(object = "transManager"): simple reporter

Examples

```
showClass("transManager")
```

transScores *obtain the top trans associations for each SNP in an smlSet*

Description

obtain the top trans associations for each SNP in an smlSet

Usage

```
transScores(smpack, snpchr = "chr1", rhs, K = 20, targdirpref = "tsco", geneApply = lapply,
  chrnames = paste("chr", as.character(1:22), sep = ""), geneRanges = NULL, snpRanges = NULL,
  radius = 2e+06, renameChrs = NULL, probesToKeep = NULL, batchsize = 200,
  genegran = 50, shortfac = 10, wrapperEndo = NULL,
  geneannopk = "illuminaHumanv1.db",
  snpannopk = "SNPlocs.Hsapiens.dbSNP.20110815", gchrpref = "",
                schrpref = "ch")

mtransScores (smpackvec, snpchr = "chr1", rhslist, K = 20, targdirpref = "multtsco",
    geneApply = lapply, chrnames = paste("chr", as.character(1:22), sep=""),
    geneRanges = NULL, snpRanges = NULL, radius = 2e+06, renameChrs=NULL,
    batchsize=200, genegran=50, probesToKeep=NULL, shortfac=10, wrapperEndo=NULL)
```

Arguments

smpack	name of package holding eset.rda providing 'ex' ExpressionSet when loaded, and holding SnpMatrix instances in inst/parts
smpackvec	vector of names of package holding eset.rda providing 'ex' ExpressionSet when loaded, and holding SnpMatrix instances in inst/parts
snpchr	name or vector of chromosome names of SNPs of interest
rhs	right hand side of snp.rhs.tests model for which expression is left hand side, e.g., covariates other than genotype
rhslist	list of right hand side of snp.rhs.tests model for which expression is left hand side, e.g., covariates other than genotype, one per element of smpackvec
K	number of most highly associated features to be retained
targdirpref	prefix of target folder name (passed to eqtlTests
geneApply	passed to eqtlTests
chrnames	names of chromosomes harboring genes that will be tested for association with genotype
geneRanges	list of GRanges-class instances containing chromosomal coordinate defined regions occupied by genes, with regions partitioned by chromosomes, and list element names as given in chrnames above
snpRanges	list of GRanges-class instances with SNP addresses
radius	radius within which an association is considered cis and therefore the corresponding test statistic is set to zero
renameChrs	passed to getSS
probesToKeep	passed to getSS
batchsize	defines batch size for ffrowapply
genegran	passed to eqtlTests
shortfac	passed to eqtlTests
wrapperEndo	a function accepting and returning an smlSet instance
gchrpref	prefix to convert chrnames into appropriate tokens for obtaining gene metadata; in future this may need to be a string transformation function
schrpref	prefix to convert chrnames into appropriate tokens for use with getSNPlocs for the SNP location information package identified in snpannopack parameter below
geneannopk	character string naming a Bioconductor .db expression chip annotation package
snpannopk	character string naming a Bioconductor SNPlocs.* SNP metadata package

Value

a list with elements

scores	an S by K ff matrix where S is number of SNPs, K is number of best features to be retained, with element s,k the kth largest score statistic among association tests computed for SNP s
inds	an S by K ff matrix with s,k element telling which element of guniv (see below) is the gene giving the kth largest score statistic for association
guniv	the vector of gene identifiers defining the universe of genes tested
snpnames	vector of SNP identifiers
call	the call used to create the result

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
## Not run:
library(GGdata)
# need to define the geneRanges and snpRanges ...
transScores("GGdata", "20", renameChrs="chr20", chrnames="chr21")

## End(Not run)
```

---

transTab *tabulate results of transScores run*

Description

tabulate results of transScores run

Usage

```
transTab(x)
```

Arguments

x	a list, as returned by slot(y, "base"), where y is a transManager instance.

Value

data.frame instance

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

---

vcf2sm *generate a SnpMatrix instance on the basis of a VCF (4.0) file*

Description

generate a SnpMatrix instance on the basis of a VCF (4.0) file.

Usage

```
vcf2sm(tbxfi, ..., gr, nmetacol)
```

Arguments

tbxfi	instance of TabixFile-class
...	not used
gr	instance of GRanges-class
nmetacol	numeric: tells number of columns used in each record as locus-level metadata

Details

This function is relevant only for diallelic SNP. If any base call is denoted '.', the associated genotype is set to missing (raw 0), even if the nonmissing call is ALT, implying at least one ALT.

Value

an instance of SnpMatrix-class

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

References

http://www.1000genomes.org/wiki/doku.php?id=1000_genomes:analysis:vcf4.0

Examples

```
# SRC: ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/release/2010_07/exon/CEU.exon.2010_03.genotypes.
 vref = system.file("vcf/CEU.exon.2010_09.genotypes.vcf.gz", package="GGtools")
 gg = GenomicRanges::GRanges(seqnames="1", IRanges::IRanges(10e6,20e6))
 vcf2sm(Rsamtools::TabixFile(vref), gr=gg, nmetacol=9L)
```

Index