GGBase

October 25, 2011

`GGbase-package` *GGbase Package Overview*

Description

GGbase Package Overview

Details

This package provides infrastructure for programming related to the genetics of gene expression. The GGtools package makes use of classes and methods defined in this package. GGdata and hmyriB36 packages use the class structures defined in this package for serialized data.

Introductory information is available from vignettes, type `openVignette()`.

Full listing of documented man pages is available in HTML view by typing `help.start()` and selecting `GGbase` package from the `Packages` menu or via `library(help="GGbase")`.

Author(s)

V. Carey

`MAFfilter` *restrict SNP in an smlSet to range of minor allele frequencies (MAF) or*

Description

restrict SNP in an smlSet to range of minor allele frequencies (MAF) or genotype frequencies

Usage

```
MAFfilter(x, lower = 0, upper = 1)
GTFfilter(x, lower = 0)
```

Arguments

x	smlSet instance
lower	numeric lower bound on minor allele frequency or genotype frequency for keeping a SNP
upper	numeric upper bound on minor allele frequency for keeping a SNP

Details

uses SnpMatrix-class summary method from snpStats

Value

revised instance of smlSet-class

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
data(smlSet.example)
sapply(smList(MAFfilter(smlSet.example, lower=.1)), dim)
sapply(smList(GTFfilter(smlSet.example, lower=.1)), dim)
```

SessionInfo-class *Class "SessionInfo" - objects to help stamp an output with information*

Description

Class "SessionInfo" – objects to help stamp an output with information on session state

Objects from the Class

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

Slots

`.S3Class`: Object of class `"character"` simple cast to allow checking

Extends

Class `"oldClass"`, by class "sessionInfo", distance 2.

Methods

No methods defined with class "SessionInfo" in the signature.

Examples

```
showClass("SessionInfo")
```

genesym-class *Class "genesym" and other casting classes*

Description

classes that help establish symbol semantics for dispatching

Objects from the Class

Objects can be created by calls of the form `new("genesym", ...)`, or by special constructor functions. As of GGBase version 3.7.1, you can use `genesym(...)`, `chrnum(...)`, `probeId(...)`, `rsid(...)`. These generally just extend character or numeric so that vector operations are straightforward, but attach type information so that methods such as `[` 'know' what they are getting.

Currently, `genesym` is used to allow HUGO symbols to be passed to `[`; `chrnum` identifies numerals or numeric constants as indices into the set of chromosomes (no chr prefix is allowed); `rsid` identifies dbSNP identifiers; `probeId` identifies a string as a microarray probe identifier.

`snpdepth` identifies a number that will be used as the number of chromosome-specific test results to be retained in any genome-wide screen

Slots

`.Data`: Object of class `"character"` ~~

Extends

Class `"character"`, from data part. Class `"vector"`, by class "character", distance 2. Class `characterORMIAME`, by class "character", distance 2.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
showClass("genesym")
genesym("CPNE1")
```

featureFilter *remove unannotated or undesired features from an smlSet instance*

Description

remove unannotated or undesired features from an smlSet instance

Usage

```
featureFilter(x, requires = c("loc", "autosomal"))
```

Arguments

x	instance of smlSet class
requires	character vector – if "loc" is present, require that a non-NA value is present in CHRLOC for each feature; if "autosomal" is present, require that CHR value is in 1:22 (presently assumes human genome)

Value

revised smlSet instance excluding features no

Author(s)

VJ Carey

Examples

```
data(smlSet.example)
dim(exprs(smlSet.example))
fff = featureFilter(smlSet.example)
dim(exprs(fff))
```

---

gwSnpScreenResult-class

Class "gwSnpScreenResult" - containers for GGtools gwSnpScreen method

---

Description

Class "gwSnpScreenResult" – container for GGtools gwSnpScreen method outputs and allied objects

Objects from the Class

Objects can be created by calls of the form `new("gwSnpScreenResult", ...)`. These will be primarily lists of inference tables (snps are rows, columns are statistics and p-values). Additional slots manage analysis metadata.

gwSnpScreenResult is intended for genome-wide analysis of expression for a single gene.

cwSnpScreenResult is intended for the restriction to a single chromosome.

multiGwSnpScreenResult is intended for analyses with multiple genes.

Because the vast majority of tests are uninformative, early filtering is important for managing object sizes. Instances of filteredGwSnpScreenResult and filteredMultiGwSnpScreenResult are created when a snpdepth parameter is used with gwSnpTests.

Slots

`.Data`: Object of class `"list"` containing inference tables (snps are rows, columns are statistics and p-values)

`gene`: Object of class `"character"` typically the HUGO symbol of the gene analyzed

`psid`: Object of class `"character"` the feature identifier of the associated microarray

`annotation`: Object of class `"character"` vector of relevant annotation package identifier names

`formula`: Object of class `"formula"` the formula used to fit the model relating expression to genotype

Extends

Class `"list"`, from data part. Class `"vector"`, by class "list", distance 2. Class `AssayData`, by class "list", distance 2.

Methods

plot and show

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
showClass("gwSnpScreenResult")
showClass("cwSnpScreenResult")
```

---

make_smlSet *create an smlSet instance from components*

Description

create an smlSet instance from components

Usage

```
make_smlSet(es, sml, organism = "Homo sapiens",
  harmonizeSamples = FALSE)
```

Arguments

`es`	ExpressionSet instance
`sml`	list of SnpMatrix instances
`organism`	string naming organism
`harmonizeSamples`	logical telling whether to intersect samples from expression and SNP data when sample sets do not coincide

Details

combines SnpMatrix instances with expression data

Value

instance of smlSet class

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
data(smlSet.example) # here we just show the mechanics from a working smlSet
es = as(smlSet.example, "ExpressionSet")
sl = smList(smlSet.example)
mm = make_smlSet(es, sl)
validObject(mm)
mm
```

---

multiCisTestResult-class

Class "multiCisTestResult"

Description

object to contain results of restricted gene-centric searches for eQTL

Objects from the Class

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

Slots

`.Data`: Object of class `"list"` – list of results of `snp.rhs.tests`

`conditions`: Object of class `"list"` – list of runtime conditions encountered

`call`: Object of class `"call"` – for auditing, the call used is saved

Extends

Class `"list"`, from data part. Class `"vector"`, by class "list", distance 2. Class `"AssayData"`, by class "list", distance 2.

Methods

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

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```
showClass("multiCisTestResult")
```

plot_EvG-methods *formal method for visualizing expression distributions vs genotype*

Description

boxplot expression vs genotype

Methods

gsym = "genesym", rsid = "rsid", sms = "smlSet" generates an annotated boxplot

multisnp methods plot_EvG2 allows specification of a second SNP rsid and shows boxplots over the cross-tabulation of the allele combinations

Examples

```
data(smlSet.example)
plot_EvG(genesym("WBP5"), rsid("rs10483083"), smlSet.example)
```

smlSet-class *Documentation on S4 class "smlSet" an eSet-derived container for*

Description

Documentation on S4 class "smlSet" an eSet-derived container for SnpMatrix lists, allowing efficient combination of SNP chip genotyping with microarray expression data, and allied classes

Objects from the Class

Objects can be created by calls of the form `new("smlSet", assayData, phenoData, featureData, experimentData, annotation, ...)`. These objects respond to interrogation on samples, expression values, SNP values, and other metadata.

Slots

`smlEnv`: Object of class `"environment"` an environment with single key `smList` pointing to a list of package snpStats SnpMatrix instances

`organism`: Object of class `"character"` informal, "Hs" recommended for human

`assayData`: Object of class `"AssayData"` intended to hold expression data coordinated with the smlEnv data

`phenoData`: Object of class `"AnnotatedDataFrame"` standard sample-level data container from eSet design

`featureData`: Object of class `"AnnotatedDataFrame"` standard feature-level metadata container, implied usage is for documenting the expresion data elements

`experimentData`: Object of class `"MIAME"` standard metadata container from Biobase eSet design

`annotation`: Object of class `"character"` vector giving the Bioconductor annotation package (.db type) for decoding expression feature identifiers.

`.__classVersion__`: Object of class `"Versions"` class version tracking metadata

Extends

Class eSet, directly. Class VersionedBiobase, by class "eSet", distance 2. Class Versioned, by class "eSet", distance 3.

Methods

smList `signature(x = "smlSet")`: retrieves the actual list of SnpMatrix entities

smlEnv `signature(x = "smlSet")`: retrieves the environment holding SnpMatrix entities

exprs `signature(x = "smlSet")`: retrieves the matrix of expression values

snps `signature(x = "smlSet", chr = "chrnum")`: retrieves the raw matrix of genotype values (SnpMatrix instance from snpStats package

`combine`: concatenates expression data and forms intersection of SNP sets

`getAlleles(smlSet, rsid)`: returns A/B notations for SNP determined by rsid

`coerce`: extracts exprs, phenoData and annotation and constructs ExpressionSet

[`signature(x = "smlSet", i = "ANY", j = "ANY", drop = "ANY")`: Quick methods for subsetting elements of smlSets have been provided.

If X is an `smlSet` instance and G is a vector of class `probeId-class`, then `X[G,]` will reduce the expression data to the probes specified in `G`.

If X is an `smlSet` instance and G is a vector of class `chrnum-class`, then `X[G,]` will reduce the SNP genotype data to the SNPs resident on chromosomes enumerated in `G`.

If X is an `smlSet` instance and G is a vector of class `rsid-class`, then `X[G,]` will reduce the SNP genotype data to the SNPs enumerated in the dbSNP id in `G`.

Note

We have included a `[` method for SnpMatrix instances that accepts an rsid instance as a column selector.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

See Also

GGtools package makes extensive use of these classes and methods.

Examples

```
showClass("smlSet")
data(smlSet.example)
smlSet.example
validObject(smlSet.example)
# workout on expression components
dim(exprs(smlSet.example))
fn = featureNames(smlSet.example)[1:10]
fn
ss2 = smlSet.example[ probeId(fn), ] # restrict exprs to set of probes
dim(exprs(ss2))
# workout on SNP components
smList(smlSet.example)
dim(smList(ss2)[[1]])
ss2[ chrnum(21), ]  # trivial restriction of SNP to a chromosome
```

```
sn = colnames(smList(ss2)[[1]])[1:20]  # get some dbSNP ids
ss3 = ss2[ rsid(sn), ]   # subset the snps
dim(smList(ss3)[[1]])
dim(smList(ss3)[["21"]]) # check names
ss3
as(snps(ss3, chrnum(21)), "character")[1:5,1:5] # generic codes
as(snps(ss3, chrnum(21)), "numeric")[1:5,1:5]  # number copies of B
as(snps(ss3, chrnum(21)), "matrix")[1:5,1:5]  # raw
```

`smlSummary` *class and function to summarize frequency information on genotypes in*

Description

generates information on sample size, minor allele frequency, specific call frequencies, and HWE test results on all SNP in an smlSet

Usage

```
smlSummary(x)
```

Arguments

x	instance of `smlSet-class`

Details

to control volume of printout a simple list extending class is defined for show method

Value

Instance of smlSummary class, which simply extends list. Each list element is a matrix of results provided by `summary,SnpMatrix-method`.

Author(s)

VJCarey <stvjc@channing.harvard.edu>

Examples

```
data(smlSet.example)
smlSummary(smlSet.example)
```

Index

∗Topic **classes**
genesym-class, 3
gwSnpScreenResult-class, 4
multiCisTestResult-class, 6
SessionInfo-class, 2
smlSet-class, 7
∗Topic **methods**
plot_EvG-methods, 7
∗Topic **models**
featureFilter, 3
MAFfilter, 1
make_smlSet, 5
smlSummary, 9
∗Topic **package**
GGbase-package, 1
[,SnpMatrix,ANY,rsid,ANY-method *(smlSet-class)*, 7
[,cwSnpScreenResult,ANY,ANY,ANY-method *(gwSnpScreenResult-class)*, 4
[,gwSnpScreenResult,ANY,ANY,ANY-method *(gwSnpScreenResult-class)*, 4
[,smlSet,ANY,ANY,ANY-method *(smlSet-class)*, 7
[,smlSet-method *(smlSet-class)*, 7

AssayData, *5, 6*

character, *3*
characterORMIAME, *3*
chrnum *(genesym-class)*, 3
chrnum,character-method *(genesym-class)*, 3
chrnum,numeric-method *(genesym-class)*, 3
chrnum-class, *8*
chrnum-class *(genesym-class)*, 3
cnumOrMissing *(genesym-class)*, 3
cnumOrMissing-class *(genesym-class)*, 3
coerce,smlSet,ExpressionSet-method *(smlSet-class)*, 7
combine,filteredMultiGwSnpScreenResult,filtere *(gwSnpScreenResult-class)*, 4
combine,multiGwSnpScreenResult,multiGwSnpScree *(gwSnpScreenResult-class)*, 4
combine,smlSet,smlSet-method *(smlSet-class)*, 7
cwSnpScreenResult-class *(gwSnpScreenResult-class)*, 4

eSet, *8*
exprs,smlSet-method *(smlSet-class)*, 7

featureFilter, 3
filteredGwSnpScreenResult-class *(gwSnpScreenResult-class)*, 4
filteredMultiGwSnpScreenResult-class *(gwSnpScreenResult-class)*, 4

genesym *(genesym-class)*, 3
genesym,character-method *(genesym-class)*, 3
genesym-class, 3
getAlleles *(smlSet-class)*, 7
getAlleles,smlSet,rsid-method *(smlSet-class)*, 7
GGbase *(GGbase-package)*, 1
GGbase-package, 1
GTFfilter *(MAFfilter)*, 1
gwSnpScreenResult *(gwSnpScreenResult-class)*, 4
gwSnpScreenResult-class, 4

list, *5, 6*

MAFfilter, 1
make_smlSet, 5
multiCisTestResult-class, 6

multiGwSnpScreenResult-class (gwSnpScreenResult-class), 4

oldClass, 2

phenoVar (genesym-class), 3
phenoVar,character-method (genesym-class), 3
phenoVar-class (genesym-class), 3
plot,cwSnpScreenResult,ANY-method (gwSnpScreenResult-class), 4
plot,cwSnpScreenResult,character-method (gwSnpScreenResult-class), 4
plot,cwSnpScreenResult,missing-method (gwSnpScreenResult-class), 4
plot,gwSnpScreenResult,ANY-method (gwSnpScreenResult-class), 4
plot,multiGwSnpScreenResult,ANY-method (gwSnpScreenResult-class), 4
plot_EvG (plot_EvG-methods), 7
plot_EvG,genesym,rsid,smlSet-method (plot_EvG-methods), 7
plot_EvG,probeId,rsid,smlSet-method (plot_EvG-methods), 7
plot_EvG-methods, 7
plot_EvG2 (plot_EvG-methods), 7
plot_EvG2,genesym,rsid,rsid,smlSet-method (plot_EvG-methods), 7
plot_EvG2,probeId,rsid,rsid,smlSet-method (plot_EvG-methods), 7
probeId (genesym-class), 3
probeId,character-method (genesym-class), 3
probeId-class, 8
probeId-class (genesym-class), 3

rsid (genesym-class), 3
rsid,character-method (genesym-class), 3
rsid,numeric-method (genesym-class), 3
rsid-class, 8
rsid-class (genesym-class), 3

SessionInfo-class, 2
show,chrnum-method (genesym-class), 3
show,cwSnpScreenResult-method (gwSnpScreenResult-class), 4
show,filteredGwSnpScreenResult-method (gwSnpScreenResult-class), 4
show,filteredMultiGwSnpScreenResult-method (gwSnpScreenResult-class), 4
show,gwSnpScreenResult-method (gwSnpScreenResult-class), 4
show,multiCisTestResult-method (multiCisTestResult-class), 6
show,multiGwSnpScreenResult-method (gwSnpScreenResult-class), 4
show,rsid-method (genesym-class), 3
show,smlSet-method (smlSet-class), 7
show,smlSummary-method (smlSummary), 9
smlEnv (smlSet-class), 7
smlEnv,smlSet-method (smlSet-class), 7
smlList (smlSet-class), 7
smlList,smlSet-method (smlSet-class), 7
smlSet (smlSet-class), 7
smlSet-class, 2, 9
smlSet-class, 7
smlSet.example (smlSet-class), 7
smlSummary, 9
smlSummary-class (smlSummary), 9
snp.rhs.tests, 6
snpdepth (genesym-class), 3
snpdepth-class (genesym-class), 3
SnpMatrix-class, 2
snpNames (smlSet-class), 7
snpNames,smlSet,chrnum-method (smlSet-class), 7
snpNames,smlSet,missing-method (smlSet-class), 7
snps (smlSet-class), 7
snps,smlSet,chrnum-method (smlSet-class), 7
summary,SnpMatrix-method, 9

updateObject,smlSet-method (smlSet-class), 7

vector, *3*, *5*, *6*
Versioned, *8*
VersionedBiobase, *8*