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", ...). and generally just extend character or numeric so that vector operations are straightforward

Currently, genesym is used to allow HUGO symbols to be passed to [:chrnum identifies numer- als 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")
genSYM("CPNE1")
Class "gwSnpScreenResult" – containers 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 were used

`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")`
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**

```r
library(GGtools)
data(hmceuB36.2021)
plot_EvG(genesym("CPNE1"), rsid("rs6060535"), hmceuB36.2021)
```

**smlSet-class**

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

**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 snpMatrix snp.matrix instances
- **chromInds**: Object of class "numeric" numeric vector indicating what chromosomes are represented in the smlEnv
- **snpNames**: Object of class "character" list of character strings naming SNPs genotyped in smList element of smlEnv
- **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 expression 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 snp.matrix entities
- **smlEnv** signature (x = "smlSet"): retrieves the environment holding snp.matrix entities
- **exprs** signature (x = "smlSet"): retrieves the matrix of expression values
- **combine**: concatenates expression data and forms intersection of SNP sets
- **getAlleles(smlSet, rsid)**: returns A/B notations for SNP determined by rsid

**Note**

We have included a [ ] method for snp.matrix 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**

```r
showClass("smlSet")
```

---

**snp.rhs.tests154  Score tests with SNP genotypes as independent variable – taken from
snpMatrix 1.5.4 while 1.5.5 version of this function is diminished in speed**

**Description**

This function fits a generalized linear model with phenotype as dependent variable and, optionally, one or more potential confounders of a phenotype-genotype association as independent variable. A series of SNPs (or small groups of SNPs) are then tested for additional association with phenotype. In order to protect against misspecification of the variance function, "robust" tests may be selected.
Usage

snp.rhs.tests154(formula, family = "binomial", link, weights, subset, data = parent.frame(),
                  snp.data, tests=NULL, robust = FALSE,
                  control=glm.test.control(maxit=20, epsilon=1.e-4, R2Max=0.98), allow.missing=0.01)

Arguments

formula  The base model formula, with phenotype as dependent variable
family   A string defining the generalized linear model family. This currently should (partially) match one of "binomial", "Poisson", "Gaussian" or "gamma" (case-insensitive)
link     A string defining the link function for the GLM. This currently should (partially) match one of "logit", "log", "identity" or "inverse". The default action is to use the "canonical" link for the family selected
data     The dataframe in which the base model is to be fitted
snp.data An object of class "snp.matrix" or "X.snp.matrix" containing the SNP data
tests    Either a vector of column names or numbers for the SNPs to be tested, or a list of short vectors defining groups of SNPs to be tested (again by name or number). The default action is to carry out all single SNP tests, but single.snp.tests will often achieve the same result much faster
weights  "Prior" weights in the generalized linear model
subset   Array defining the subset of rows of data to use
robust   If TRUE, robust tests will be carried out
control  An object giving parameters for the IRLS algorithm fitting of the base model and for the acceptable aliasing amongst new terms to be tested. See code glm.test.control
allow.missing The maximum proportion of SNP genotype that can be missing before it becomes necessary to refit the base model

Details

The tests used are asymptotic chi-squared tests based on the vector of first and second derivatives of the log-likelihood with respect to the parameters of the additional model. The "robust" form is a generalized score test in the sense discussed by Boos(1992). The "base" model is first fitted, and a score test is performed for addition of one or more SNP genotypes to the model. Homozygous SNP genotypes are coded 0 or 2 and heterozygous genotypes are coded 1. For SNPs on the X chromosome, males are coded as homozygous females. For X SNPs, it will often be appropriate to include sex of subject in the base model (this is not done automatically).

If a data argument is supplied, the snp.data and data objects are aligned by rowname. Otherwise all variables in the model formulae are assumed to be stored in the same order as the columns of the snp.data object.

Value

A data frame containing, for each SNP,

Chi.squared  The value of the chi-squared test statistic
Df           The corresponding degrees of freedom
\textbf{Df.residual} \hspace{1em} The residual degrees of freedom for the base model; \textit{i.e.} the number of observations minus the number of parameters fitted

For the binomial family model, the base model can, in some circumstances, lead to perfect prediction of some observations (\textit{i.e.} fitted probabilities of 0 or 1). These observations are ignored in subsequent calculations; in particular they are not counted in the residual degrees of freedom. Similarly for Poisson means fitted exactly to zero.

\textbf{Note}

A factor (or several factors) may be included as arguments to the function \texttt{strata(...)} in the formula. This fits all interactions of the factors so included, but leads to faster computation than fitting these in the normal way. Additionally, a \texttt{cluster(...)} call may be included in the base model formula. This identifies clusters of potentially correlated observations (e.g. for members of the same family); in this case, an appropriate robust estimate of the variance of the score test is used.

\textbf{Author(s)}

David Clayton \langle david.clayton@cimr.cam.ac.uk \rangle

\textbf{References}


\textbf{See Also}

\texttt{single.snp.tests, snp.lhs.tests, snp.matrix-class, X.snp.matrix-class}

\textbf{Examples}

\begin{verbatim}
library(snpMatrix)
data(testdata)
slt3 <- snp.rhs.tests154(cc~strata(region), family="binomial",
data=subject.data, snp.data= Autosomes, tests=1:10)
print(slt3)
\end{verbatim}

\begin{verbatim}
\hline
\textbf{snpLocs.Hs} & \textit{SNP location accessor} \\
\hline
\end{verbatim}

\textbf{Description}

\textit{SNP location accessor}

\textbf{Usage}

\texttt{snpLocs.Hs(cnum, rsid)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{cnum} \hspace{1em} chnum instance indicating chromosome
  \item \texttt{rsid} \hspace{1em} rsid instance giving dbSNP ids for snps of interest
\end{itemize}
snpsNear

Details

The SNPlocs.Hsapiens.dbSNP.* package is curated by Bioconductor and maintains location and allele information on SNP.

The hsSnpLocs environment is available through data(hsSnpLocs), and contains a unified representation of the information in the SNPlocs package (which is only available through various chromosome-specific calls).

Value

two-row matrix – top row is numeric suffix of dbSNP ids, bottom row is location, genome-wide if chrnum is missing

Author(s)

Vince Carey <stvjc@channing.harvard.edu>

Examples

library(GGtools)
data(hmceuB36.2021)
nn21 = colnames(smList(hmceuB36.2021)[['21']])
length(nn21)
ss = snpLocs.Hs(chrnum(21), rsid(nn21))
dim(ss)
# notes that the locations available are not completely
# congruent with those asserted in HapMap data

snpsNear

obtain list of rs numbers for snps near a gene

Description

obtain list of rs numbers for snps near a gene

Usage

snpsNear(sym, radius=1e+05, chrnum)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sym</td>
<td>instance of genesym class [e.g., use genesym(string) for gene 'string'], or of rsid class, or of numeric class</td>
</tr>
<tr>
<td>radius</td>
<td>number of base-pairs in each direction to look</td>
</tr>
<tr>
<td>chrnum</td>
<td>chrnum instance .. optional</td>
</tr>
</tbody>
</table>

Details

simple arithmetic based on output of snpLocs.Hs
Value

character vector of rsxxxxx, dbSNP id, according to locations from SNPlocs.Hsapiens.dbSNP.20071016 package, as transferred to snpLocs.Hs resource in GGBase

Note

first invocation can take longer than subsequent, if snpLocs.Hs has not been invoked previously

Author(s)

Vince Carey <stvjc@channing.harvard.edu>

Examples

```r
nearc = snpsNear(genesym("CPNE1"), 10000, chrnum(20))
library(GGtools)
data(hmceuB36.2021)
ss = smList(hmceuB36.2021)[[1]]
# following calculation requires new "[" for j an instance of rsid
clo = ss[ , rsid(snpsNear(rsid("rs6060535"), rad=1500, chrnum(20))) ]
clo
```
Index

**Topic classes**
- genesym-class, 1
- gwSnpScreenResult-class, 2
- smlSet-class, 3

**Topic htest**
- snp.rhs.tests154, 4

**Topic methods**
- plot_EvG-methods, 3

**Topic models**
- snpLocs.Hs, 6
- snpsNear, 7
  - [,cwSnpScreenResult,ANY,ANY,ANY-method (gwSnpScreenResult-class), 2
  - [,smlSet,ANY,ANY,ANY-method (smlSet-class), 3
  - [,snp.matrix,ANY,rsid,ANY-method (smlSet-class), 3
- AssayData, 2
- character, 1
- characterORMIAME, 1
- chrnum (genesym-class), 1
- chrnum, character-method (genesym-class), 1
- chrnum, numeric-method (genesym-class), 1
- chrnum-class (genesym-class), 1
- cnnumOrMissing (genesym-class), 1
- cnnumOrMissing-class (genesym-class), 1
- combine, smlSet, smlSet-method (smlSet-class), 3
- cwSnpScreenResult-class (gwSnpScreenResult-class), 2
- eSet, 4
- exprs, smlSet-method (smlSet-class), 3
- filteredGwSnpScreenResult-class (gwSnpScreenResult-class), 2
- filteredMultiGwSnpScreenResult-class (gwSnpScreenResult-class), 2
- genesym (genesym-class), 1
- genesym, character-method (genesym-class), 1
- genesym-class, 1
- getAlleles (smlSet-class), 3
- getAlleles, smlSet, rsid-method (smlSet-class), 3
- getSpnLocs (snpLocs.Hs), 6
- getSpnLocs, smlSet, chrnum-method (snpLocs.Hs), 6
- getSpnLocs, smlSet, missing-method (snpLocs.Hs), 6
- glm.test.control, 5
- gwSnpScreenResult-class, 2
- hsSnpsLocs (snpLocs.Hs), 6
- list, 2
- multiGwSnpScreenResult-class (gwSnpScreenResult-class), 2
- plot, cwSnpScreenResult, ANY-method (gwSnpScreenResult-class), 2
- plot, gwSnpScreenResult, ANY-method (gwSnpScreenResult-class), 2
- plot, multiGwSnpScreenResult, ANY-method (gwSnpScreenResult-class), 2
- plot_EvG (plot_EvG-methods), 3
- plot_EvG, genesym, rsid, smlSet-method (plot_EvG-methods), 3
- plot_EvG, probeId, rsid, smlSet-method (plot_EvG-methods), 3
- plot_EvG-methods, 3
- plot_EvG2 (plot_EvG-methods), 3
- plot_EvG2, genesym, rsid, rsid, smlSet-method (plot_EvG-methods), 3
plot_EvG2, probeId, rsid, rsid, smlSet-method
(plot_EvG-methods), 3
probeId(genesym-class), 1
probeId, character-method (genesym-class), 1
probeId-class(genesym-class), 1
rsid(genesym-class), 1
rsid, character-method (genesym-class), 1
rsid, numeric-method (genesym-class), 1
rsid-class(genesym-class), 1

show, chrnum-method (genesym-class), 1
show, cwSnspScreenResult-method (gwSnspScreenResult-class), 2
show, filteredGwSnspScreenResult-method (gwSnspScreenResult-class), 2
show, filteredMultiGwSnspScreenResult-method (gwSnspScreenResult-class), 2
show, gwSnspScreenResult-method (gwSnspScreenResult-class), 2
show, multiGwSnspScreenResult-method (gwSnspScreenResult-class), 2
show, rsid-method(genesym-class), 1
show, smlSet-method (smlSet-class), 3
show, snpLocs-method(snpLocs.Hs), 6

single.snp.tests, 5, 6
smlEnv(smlSet-class), 3
smlEnv, smlSet-method (smlSet-class), 3
smlList(smlSet-class), 3
smlList, smlSet-method (smlSet-class), 3
smlSet-class, 3
snp.lhs.tests, 6
snp.matrix-class, 6
snp.rhs.tests154, 4
snpdepth(genesym-class), 1
snpdepth-class (genesym-class), 1
snpLocs-class(snpLocs.Hs), 6
snpLocs.Hs, 6