

# Package ‘GGBase’

March 18, 2019

**Title** GGBase infrastructure for genetics of gene expression package  
GGtools

**Version** 3.45.0

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

**Description** infrastructure

**Suggests** GGtools, illuminaHumanv1.db, knitr

**Depends** R (>= 2.14), methods, snpStats

**Imports** limma, genefilter, Biobase, BiocGenerics, S4Vectors, IRanges,  
Matrix, AnnotationDbi, digest, GenomicRanges,  
SummarizedExperiment

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

**License** Artistic-2.0

**LazyLoad** yes

**VignetteBuilder** knitr

**biocViews** Genetics, Infrastructure

**Collate** casting.R smlSet.R filters.R plot\_EvG.R externalize.R getSS.R  
make\_smlSet.R permEx.R

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

**git\_branch** master

**git\_last\_commit** 9942e68

**git\_last\_commit\_date** 2018-10-30

**Date/Publication** 2019-03-17

## R topics documented:

clipPCs . . . . .	2
externalize . . . . .	3
MAFilter . . . . .	4
make_smlSet . . . . .	5
plot_EvG . . . . .	6
rsid-class . . . . .	7
smlSet-class . . . . .	8
<b>Index</b>	<b>10</b>

clipPCs

*transformations of expression data in smlSet instances***Description**

transformations of expression data in smlSet instances or assay data in SummarizedExperiment

**Usage**

```
clipPCs(x, inds2drop, center = TRUE)
```

```
regressOut(sms, rhs, ...)
```

**Arguments**

x	instance of <a href="#">smlSet</a> or <a href="#">RangedSummarizedExperiment</a>
sms	instance of <a href="#">smlSet</a> or <a href="#">RangedSummarizedExperiment</a>
inds2drop	Vector of PCs to be eliminated by setting the associated diagonal elements in the SVD to zero before recomposing the matrix of expression values. If the value 0 is present in inds2drop, the smlSet is returned unchanged, with a message.
center	logical, passed to <a href="#">prcomp</a>
rhs	formula fragment (no dependent variable) used to form residuals in a reexpression of the expression matrix; variable bindings found in pData of an ExpressionSet or colData of a SummarizedExperiment
...	arguments passed to <a href="#">lmFit</a>

**Details**

clipPCs is an operation on the  $n \times p$  transposed matrix  $X$  of expression data. The singular value decomposition  $X = UDV^t$  is formed, the diagonal elements of  $D$  corresponding to inds2drop are set to zero yielding the diagonal matrix  $E$ , and then  $Y = UEV^t$  is computed and transposed to replace the expression data.

regressOut obtains residuals after genewise regression of expression on the design matrix specified by the rhs; [lmFit](#) is used to compute coefficients, linear predictions and residuals.

**Value**

an instance of [smlSet](#)

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**References**

The use of PCA-based adjustments to remove mass extraneous effects from expression matrices has been criticized in work of Oliver Stegle and Jeffrey Leek, who offer Bayesian PEER and SVA respectively as alternative solutions. The PCA-based method seems to have reasonable effectiveness in examples worked with GGdata.

**Examples**

```
## Not run: # this would induce cyclic dependency, but should
# run manually
if ("GGtools" %in% installed.packages()[,1]) {
  require("GGtools")
  c20 = getSS("GGtools", "20")
  t1 = gwSnpTests(genesym("CPNE1")~male, c20, chrnum("20"))
  topSnps(t1)
  c20c = clipPCs(c20, 1:10)
  t2 = gwSnpTests(genesym("CPNE1")~male, c20c, chrnum("20"))
  topSnps(t2)
}

## End(Not run)
```

---

externalize	<i>create a package with a decomposed smlSet instance from a unified smlSet instance, to reduce memory footprints</i>
-------------	---

---

**Description**

create a package with a decomposed smlSet instance from a unified smlSet instance, to reduce memory footprints

**Usage**

```
externalize(smlSet, packname,
  author = "Replace Me <auth@a.b.com>",
  maintainer = "Replace Me <repl@a.b.com>")

allsnps(packname)
```

**Arguments**

smlSet	an smlSet instance to be regarded as a source or template for a package that can be used with greater efficiency of access to genotype contents
packname	name of the package to be generated
author	string to be used in DESCRIPTION file of generated package
maintainer	string to be used in DESCRIPTION file of generated package

**Details**

The genotype content of the input smlSet is separated into separate RDA files in the inst/parts folder of the package to be generated. The ExpressionSet element of the input smlSet is stored as object ex in file eset.rda in the data folder of the package to be generated.

**Value**

creates folder structure and metadata for an installable R package

**Note**

allsnps() gives the list of vectors of snpnames in use in the package

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
## Not run:
setwd(tempdir())
sms = getSS("GGtools", c("20", "21"))
externalize(sms, "demopack")
dir()
dir("demopack")

## End(Not run)
```

---

MAFfilter

*Filter genotype contents of an smlSet according to certain SNP allele frequency features.*

---

**Description**

Filter genotype contents of an smlSet according to certain SNP allele frequency features.

**Usage**

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

```
GTFfilter(x, lower = 0)
```

```
dropMonomorphies(sms)
```

**Arguments**

x	<a href="#">smlSet-class</a> instance
sms	<a href="#">smlSet-class</a> instance
lower	lower bound on MAF or GTF to allow retention of associated locus
upper	upper bound on MAF or GTF to allow retention of associated locus

**Details**

uses [col.summary](#) to compute MAF or GTF. dropMonomorphies also uses col.summary.

**Value**

[smlSet-class](#) instance

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```

if (file.exists(system.file("parts/20.rda", package="GGtools"))) {
  c20 = getSS("GGtools", "20")
  c20
  c20f = MAFFilter(c20, lower=.05)
  c20f
}

```

---

make\_smlSet

*construct an smlSet instance from existing resources*


---

**Description**

construct an smlSet instance from existing resources, either using ExpressionSet and SnpMatrix instances, or a suitably structured package

**Usage**

```

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

getSS(packname, chrs, renameChrs, probesToKeep=NULL, exFilter=function(x)x,
      wrapperEndo=NULL, checkValid=TRUE)

```

**Arguments**

es	instance of <a href="#">ExpressionSet-class</a>
sml	named list of <a href="#">SnpMatrix-class</a> instances
organism	conventional token for species
harmonizeSamples	logical indicating whether steps should be taken to be sure that the components represent identical sets of samples. can be time consuming so defaults to FALSE, in which case the user must be sure that the genotype and expression components are compatible in terms of sample content
packname	string naming the installed package from which expression and genotype data will be acquired
chrs	character vector naming the prefixes of genotype files to be used in the resulting smlSet instance; see notes below
renameChrs	character vector of same length as chrs specifying one-to-one renaming operation for genotype components in smlSet
probesToKeep	character vector identifying probes to be retained in the constructed smlSet – to be deprecated in favor of exFilter
exFilter	function that should accept and return ExpressionSet, will be executed just after probesToKeep filter if present
wrapperEndo	function that accepts and returns an smlSet instance, allowing any sort of transformation of contents acquired with getSS, executed after all assembly and filtering completed.
checkValid	logical, if TRUE, function will fail if created smlSet instance does not pass validObject()

**Details**

Packages that work with `getSS` can be created out of existing `smlSet` instances using [externalize](#).

**Value**

Instance of `smlSet-class`.

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
if ("GGtools" %in% installed.packages()[,1]) {
  s20 = getSS("GGtools", "20", renameChrs="chr20")
  s20
  make_smlSet( as(s20, "ExpressionSet"), smlist(s20) )
}
```

---

plot\_EvG

*display the association between expression values and genotypes in an smlSet instance*

---

**Description**

display the association between expression values and genotypes in an `smlSet` instance

**Usage**

```
plot_EvG(gsym, rsid, sms, ...)
```

**Arguments**

<code>gsym</code>	instance of class <code>genesym</code> or <code>probeId</code> , casting a string that names a gene (which will be looked up using the <code>annotation</code> slot of <code>sms</code> ) or a probe which must be resident on the array underlying the expression content of <code>sms</code>
<code>rsid</code>	instance of class <code>rsid</code> naming a SNP with genotype values given among the columns of the <code>smlist</code> components of <code>sms</code>
<code>sms</code>	an instance of <code>smlSet</code>
<code>...</code>	additional parameter to <code>plot</code>

**Details**

When the genotype is categorical, will use boxplots; when genotype has been imputed and includes expected allele counts, will use a scatterplot.

**Value**

a plot is rendered on the current display

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
if ("GGtools" %in% installed.packages()[,1]) {
  s20 = getSS("GGtools", "20")
  plot_EvG(genesym("CPNE1"), rsid("rs6060535"), s20)
}
```

---

rsid-class

Class "rsid"

---

**Description**

rsid(), probeId(), and genesym() are basic casting methods that assign a type to a token.

**Objects from the Class**

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

**Slots**

**.Data:** Object of class "character" that holds the content to which a type is associated by the method.

**Extends**

Class "[character](#)", from data part. Class "[vector](#)", by class "character", distance 2. Class "[data.frameRowLabels](#)", by class "character", distance 2. Class "[SuperClassMethod](#)", by class "character", distance 2. Class "[characterORconnection](#)", by class "character", distance 2. Class "[characterORMIAME](#)", by class "character", distance 2. Class "[atomicVector](#)", by class "character", distance 2. Class "[index](#)", by class "character", distance 2.

**Methods**

No methods defined with class "probeId" or "genesym" in the signature. However [ with a [smlSet-class](#) argument will dispatch differently if instances of these classes are supplied. Various GGtools reporting functions will use "rsid" for dispatch.

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
showClass("rsid")
```

---

smlSet-class	Class "smlSet"
--------------	----------------

---

### Description

Integrative container for expression plus genotype data. Genotypes are stored in an efficient format defined in the `snpStats` package.

### Objects from the Class

Objects can be created by calls of the form `new("smlSet", assayData, phenoData, featureData, experimentData, ...)`. The `make_smlSet` function can also be used to build `smlSet` instances.

### Slots

**smlEnv**: Object of class "environment" that has a key `smlList` element, to which a list of [SnpMatrix-class](#) instances is bound.

**annotation**: Object of class "character", describes `featureNames` component in terms of the name of the annotation package that can be used to decode expression probe names.

**organism**: Object of class "character", a conventional string.

**assayData**: Object of class "AssayData", manages the expression data. See [AssayData-class](#).

**phenoData**: Object of class "AnnotatedDataFrame", manages sample level data. See [AnnotatedDataFrame-class](#).

**featureData**: Object of class "AnnotatedDataFrame", manages metadata on expression probes.

**experimentData**: Object of class "MIAXE", manages metadata on experiment as a whole. See [MIAXE-class](#).

**protocolData**: Object of class "AnnotatedDataFrame", additional storage for experimental protocol description. See [eSet-class](#).

**.\_\_classVersion\_\_**: Object of class "Versions"; internal management of class version.

### Extends

Class "[eSet](#)", directly. Class "[VersionedBiobase](#)", by class "eSet", distance 2. Class "[Versioned](#)", by class "eSet", distance 3.

### Methods

[ `signature(x = "smlSet", i = "ANY", j = "ANY", drop = "ANY")`]: will restrict the content of the `smlSet` instance according to features of the arguments supplied. If `x` is numeric or a [probeId-class](#), the expression content will be restricted. If `y` is numeric or character, samples will be restricted accordingly.

**combine** `signature(x = "smlSet", y = "smlSet")`: This method attempts to amalgamate two `smlSet` instances in the appropriate way – assuming that samples are disjoint.

**nsFilter** `signature(eset = "smlSet")`: executes `genefilter`'s method (see [nsFilter](#), [ExpressionSet-method](#)) on the expression content, and then propagates the additional genotype and sample level content unchanged.

**smlList** `signature(x = "smlSet")`: retrieves the list of `SnpMatrix` instances defining the genotype content.

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

**permEx** signature(*sms* = "sm1Set"): uses `sample()` to generate a permutation of sample indices so that expression data are permuted against genotype data, but original unpermuted sample identifiers are preserved; this is necessitated by the rematching behavior of [snp.rhs.tests](#).

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
showClass("sm1Set")
```

# Index

- \*Topic **classes**
  - rsid-class, 7
  - smlSet-class, 8
- \*Topic **models**
  - clipPCs, 2
  - externalize, 3
  - MAfilter, 4
  - make\_smlSet, 5
  - plot\_EvG, 6
- [, smlSet, ANY, ANY, ANY-method (smlSet-class), 8
  
- allsnps (externalize), 3
- atomicVector, 7
  
- casting (rsid-class), 7
- character, 7
- characterORconnection, 7
- characterORMIAME, 7
- clipPCs, 2
- clipPCs, RangedSummarizedExperiment, numeric, logical-method (clipPCs), 2
- clipPCs, RangedSummarizedExperiment, numeric, missing-method (clipPCs), 2
- clipPCs, smlSet, numeric, logical-method (clipPCs), 2
- clipPCs, smlSet, numeric, missing-method (clipPCs), 2
- clipPCs, SummarizedExperiment, numeric, logical-method (clipPCs), 2
- clipPCs, SummarizedExperiment, numeric, missing-method (clipPCs), 2
- col.summary, 4
- combine, smlSet, smlSet-method (smlSet-class), 8
  
- data.frameRowLabels, 7
- dropMonomorphies (MAfilter), 4
  
- eSet, 8
- exprs, smlSet-method (smlSet-class), 8
- externalize, 3, 6
  
- genesym, 6
- genesym (rsid-class), 7
- genesym, character-method (rsid-class), 7
- genesym-class (rsid-class), 7
- getSS (make\_smlSet), 5
- GTFfilter (MAfilter), 4
  
- index, 7
  
- lmFit, 2
  
- MAfilter, 4
- make\_smlSet, 5
  
- nsFilter, smlSet-method (smlSet-class), 8
  
- permEx (smlSet-class), 8
- permEx, smlSet-method (smlSet-class), 8
- plot\_EvG, 6
- plot\_EvG, genesym, rsid, smlSet-method (plot\_EvG), 6
- plot\_EvG, probeId, rsid, smlSet-method (plot\_EvG), 6
- probeId, 6
- probeId (rsid-class), 7
- probeId, character-method (rsid-class), 7
- probeId-class (rsid-class), 7
  
- RangedSummarizedExperiment, 2
- regressOut (clipPCs), 2
- rsid, 6
- rsid (rsid-class), 7
- rsid, character-method (rsid-class), 7
- rsid, numeric-method (rsid-class), 7
- rsid-class, 7
  
- smlList (smlSet-class), 8
- smlList, smlSet-method (smlSet-class), 8
- smlSet, 2, 6
- smlSet-class, 8
- snp.rhs.tests, 9
- SuperClassMethod, 7
  
- vector, 7
- Versioned, 8
- VersionedBiobase, 8