# Package 'splicegear'

October 18, 2017

**Title** splicegear

**Version** 1.48.0

**Author** Laurent Gautier <laurent@cbs.dtu.dk>

**Description** A set of tools to work with alternative splicing

**Maintainer** Laurent Gautier <laurent@cbs.dtu.dk>

**License** LGPL

**Depends** R (>= 2.6.0), methods, Biobase(>= 2.5.5)

**Imports** annotate, Biobase, graphics, grDevices, grid, methods, utils, XML

**LazyLoad** yes

**biocViews** Infrastructure, Transcription

**NeedsCompilation** no

## R topics documented:

`as.data.frame.SpliceExprSet` *SpliceExprSet object to data.frame converter*

## Description

Converts a SpliceExprSet object to a data.frame.

## Usage

```
## S3 method for class 'SpliceSites'
as.data.frame(x, row.names = NA, optional = NA, ...)

## S3 method for class 'SpliceExprSet'
as.data.frame(x, row.names = NA, optional = NA, ...)
```

## Arguments

| | |
|---|---|
| `x` | object `SpliceSites-class` or `SpliceExprSet-class`. |
| `row.names` | `NULL` or a character vector giving the row names for the data frame. Missing values are not allowed. |
| `optional` | logical. If `TRUE`, setting row names is optional. |
| `...` | currently ignored. |

## Details

Data are traditionally stored in objects of class `data.frame`. This function links the object-oriented design of the package with the large amount of functions working on data.frames.

## Value

A `data.frame`. For both functions the first column names are `begin`, `end`, `isintypeI`, `isintypeII`, `exprs` and `genenames`. In the case of `as.data.frame.SpliceExprSet`, the next variable names will the ones in the `AnnotatedDataFrame-class` attribute of the `ExpressionSet-class` object belonging the `SpliceExprSet-class`. The last variable names will be the ones in the slot `info` of the `Probes-class` object.

## Author(s)

Laurent Gautier

## Examples

```
data(spliceset)

dataf <- as.data.frame(spliceset)

lm.panel <- function(x, y, ...) {
                                  points(x,y,...)
                                  p.lm <- lm(y~x); abline(p.lm)
                                }
```

```
## probe intensity values conditioned by the position of the probes on
## the mRNA
coplot(log(exprs) ~ Material | begin, data=dataf, panel=lm.panel)
```

---

`barplot.SpliceSites` *barplot for SpliceSites*

---

## Description

Displays a barplot of the associated AnnotatedDataFrame.

## Usage

```
## S3 method for class 'SpliceSites'
barplot(height, type.as = c("typeI", "typeII", "all"),
                    info = "tissue", ...)
```

## Arguments

| | |
|---|---|
| `height` | object of class `SpliceSites-class`. |
| `type.as` | the type of alternative splicing (see `SpliteSites-class` for further details). |
| `info` | the name of the covariate in the AnnotatedDataFrame (see details). |
| `...` | optional parameters to be passed to the underlying function `barplot`. |

## Details

When the objects are built from the XML format we propose as an exchange, the parameter `info` can at least take the values `"tissue"` and `"histology"`. One can refer to the slots `spsiteIpos.pData` and `spsiteIIpos.pData` to know what are the possible choices.

## Value

See the value returned by the function `barplot`.

## See Also

`SpliceSites-class`, `barplot`

## Examples

```
data(spsites)
barplot(spsites)
```

buildSpliceSites *Functions to query PALSdb*

## Description

Functions to make a query on PALSdb, and build objects from the result of a query.

## Usage

```
queryPALSdb(query, disp = c("data", "browser"),
          field = c("keyword", "ug.id", "gb.id", "human.cytoband", "mouse.cytoband", "cluster_cour
            species = c("human", "mouse"),
            e.value = "1e-1",
            ident.threshold = c("90% 50b", "95% 50b", "90% 45b"))

getPALSdbURL(query, disp = c("data", "browser"),
          field = c("keyword", "ug.id", "gb.id", "human.cytoband", "mouse.cytoband", "cluster_cour
            species = c("human", "mouse"),
            e.value = "1e-1",
            ident.threshold = c("90% 50b", "95% 50b", "90% 45b"))

buildSpliceSites(xml, verbose=TRUE)
```

## Arguments

| | |
|---|---|
| query | query string |
| xml | an object of class XML (as returned by xmlTreeParse) |
| disp | (idem genbank and pubmed) |
| field | The field on which the query will be based |
| species | the specie to work with |
| e.value | E-value |
| ident.threshold | threshold for matching sequences |
| verbose | verbose output. |

## Details

queryPALSdb returns an an object of class XML when disp = "data".

The function buildSpliceSites constructs SpliceSites objects from the XML data. The variables in the slots spsiteIpos.pData and spsiteIIpos.pData are at least tissue (tissue information), histology and site (site numbering).

## Value

An object of class XML for queryPALSdb, an URL for getPALSdbURL or a list of objects of class SpliceSites.

## Author(s)

laurent@cbs.dtu.dk

## References

"Standardazed output for putative alternative splicing: application to the study of splice variants with microarrays", Gautier L. et al., 2003, manuscript in preparation.

## See Also

queryPALSdb

## Examples

```
library(XML)

filename <- system.file("extdata", "example.xml", package="splicegear")

xml <- xmlTreeParse(filename, asTree=TRUE)

spsites <- buildSpliceSites(xml)
```

---

getRelSignStrength *functions to perform SPLICE*

---

## Description

Implementations of the SPLICE algorithm

## Usage

```
getRelSignStrength(x, tissue = as.factor(1:ncol(x)), fun = mean, nipt = 30, nitt = 30, ...)

getFinalRatio(x, tissue=as.factor(1:ncol(x)), fun=mean, ...)
```

## Arguments

| | |
|---|---|
| x | a matrix. One probe per line, one column per sample. Typically this would be the slot exprs of an instance of class ExprSet. |
| tissue | a covariate (factor) about the samples. |
| fun | a function to obtain a summary value (mean by default) |
| nipt | see reference. |
| nitt | see reference. |
| ... | optional parameters for the function fun |

## Details

getFinalRatio will call getRelSignStrength. The values are log-transformed. It is probably a good idea to avoid feeding function with values that are already on log scale.

## Value

A matrix of the same dimension than the input x, holding 'RSS' (Relative Signal Strength) or 'final ratios' respectively, as described in the reference. Two attributes nip and nit are attached the returned matrix.

### Author(s)

laurent@cbs.dtu.dk

### References

Genome Research (2001), Hu et. al., vol. 11, p.1244

### Examples

```
data(spliceset)

## The intensity values in the example are log-transformed.
## Undo by taking the exponential
exprs(spliceset) <- exp(exprs(spliceset))

## Re-order the rows of different slots to have the probes sorted by
## position
spliceset <- sort.SpliceExprSet(spliceset)
## extract the expression matrix
expr.m <- exprs(spliceset)
fr <- getFinalRatio(expr.m, tissue=pData(spliceset@eset)[[1]])
```

---

## grid.expand.gp *convenience functions for grid*

---

### Description

Convenience function to use the package grid

### Usage

```
grid.expand.gp(n, parlist = list())
grid.make.numeric2npc(x, xlim=NULL, lower.blank=0, upper.blank=0)
```

### Arguments

| | |
|---|---|
| n | number of parameters |
| parlist | list of parameters |
| x | numeric value |
| xlim | range for Xs |
| lower.blank, upper.blank | size for margins |

### Details

call the function gpar on the list of parameters.

### Value

Function used for its side effect.

## See Also

lattice

---

`grid.plot.Probes` *Plot splicegear objects*

---

## Description

Plot objects defined in the package splicegear

## Usage

```
grid.plot.Probes(x, col = "black", add = FALSE, probepos.yscale = NULL,
              xlim = NULL, vp = NULL, ...)

grid.plot.SpliceSites(x, col.typeI = "orange", col.typeI.window = "yellow",
    col.typeII = "red", add = FALSE, ylim = NULL, vp = NULL,
    ...)

grid.plot.SpliceExprSet(x, probes.opt = list(), expr.opt = list(col = NA, lty = 1:6),
    fig.xratio = c(2, 1), fig.yratio = c(2, 1), probepos.yscale = NULL,
    ylim = NULL, ...)
```

## Arguments

| | |
|---|---|
| `x` | object of Probes-class, SpliceSites-class or SpliceExprSet-class |
| `add` | add to an existing plot |
| `col` | color(s) for the probes (recycled if necessary). |
| `col.typeI` | color(s) for the type I spliced out exons |
| `col.typeI.window` | background color for the type I spliced out 'windows' |
| `col.typeII` | color for the type II splicing events |
| `expr.opt` | list of options to plot expression values |
| `probepos.yscale` | specify coordinates on the y-axis for the probes. |
| `probes.opt` | options to plot the probes |
| `fig.xratio` | ratio for the left and right parts of the plot |
| `fig.yratio` | ratio for the upper and lower parts of the plot |
| `vp` | a viewport (grid package stuff) |
| `xlim` | range for the x-axis (see `plot`). |
| `ylim` | range for the y-axis |
| `...` | optional parameters |

## Details

The 'type I'/'type II' thing is described in the references found in the help files for `plot.SpliceSites`.

## Value

These functions are mainly used for their side effects. `grid.plot.SpliceSites` returns the range for the y-axis when needed.

## See Also

plot.SpliceSites, plot.Probes

## Examples

```
## plot splice sites
data(spsites)
grid.plot(spsites)

## plot probes
data(probes)
grid.plot(probes)

## combined plot
grid.plot(probes, spsites)
```

---

# isProbeOnSpliceSite *Check the presence of probes on certain exons*

---

## Description

Return whether the probes are located on exons involved in (putative) alternative splicing or not.

## Usage

```
isProbeOnSpliceSite(probes, spSites)
## isSpliceSiteOnProbe is not yet implemented
```

## Arguments

| | |
|---|---|
| `probes` | object of class `Probes` |
| `spSites` | object of class `spliceSites` |

## Value

The returned value in a list of two vectors of mode `logical` of the same length:

| | |
|---|---|
| `isintypeI` | whether the probes are in a 'type I' region or not. |
| `isintypeII` | whether the probe are in a 'type II' region or not. |

## Author(s)

Laurent

## References

For details about 'type I' and 'type II', please refer to Huang Y.-H and Chen Y.-T and Lai J.-J. and Yang S.-T. and Yang U.-C., PALSdb: Putative Alternative Splicing database, Nucleic Acids Research, 2002, pages 186-190

---

`matchprobes2Probes` *create Probes object from matchprobes results*

---

## Description

Create Probes object from results the results of the function matchprobes (in the package 'matchprobes').

## Usage

```
matchprobes2Probes(mpo, probes.length, names = NULL)
```

## Arguments

| | |
|---|---|
| `mpo` | `Probes-class` object |
| `probes.length` | Length for the probes (see details). |
| `names` | names for the elements in the list returned. |

## Details

Currently only probes of unique length are assumed. In the case of Affymetrix chips, 25 base pairs is the value you probably want.

## Value

A `list` of `Probes-class` objects.

## References

http://www.cbs.dtu.dk/laurent/download/splicegear/

## See Also

`Probes-class`, the package `matchprobes`

`plot.SpliceExprSet` *plot a SpliceExprSet*

## Description

Plot a object of class `SpliceExprSet`

## Usage

```
## S3 method for class 'SpliceExprSet'
plot(x,
                    probes.opt = list(), expr.opt = list(col = NA, lty = 1:6),
                    fig.xratio = c(2, 1), fig.yratio = c(2, 1),
                    probepos.yscale = NULL, ylim,
                    ...)
```

## Arguments

| | |
|---|---|
| `x` | a `SpliceExprSet-class` |
| `probes.opt` | optional parameters to be passed for the plotting of the `Probes-class` |
| `expr.opt` | optional parameters to be passed for the plotting of the `ExpressionSet-class` |
| `fig.xratio` | ratio between the left and right parts of the plot |
| `fig.yratio` | ratio between the upper and lower parts of the plot |
| `probepos.yscale` | enforce 'y' positions for the probes. |
| `ylim` | range for the y-axis |
| `...` | optional parameters to be passed to the function `plot` |

## Details

The argument `probepos.yscale` can be used to scale probes according to their position on the reference sequence, as shown in the last example below.

## Value

function used for its side-effect(s).

## Author(s)

laurent

## See Also

`SpliceExprSet-class`

## Examples

```
data(spliceset)


levels(pData(spliceset@eset)$Material)
## Liver, Mix and SNB19
cl.mat <- c("red", "yellow","blue")[as.integer(pData(spliceset@eset)$Material)]
## colored in red, yellow and blue respectively
plot(spliceset, expr.opt = list(col = cl.mat, log = "x"))

## sort
spliceset <- sort.SpliceExprSet(spliceset)
begin.pos <- spliceset@probes@pos[, 1]
plot(spliceset, expr.opt = list(col=cl.mat), probepos.yscale = begin.pos)
```

---

`plot.SpliceSites` *plot a SpliceSites object*

---

## Description

plot objects.

## Usage

```
## S3 method for class 'Probes'
plot(x, col="black",
            xlab = "sequence", ylab = "probes",
            add=FALSE, probepos.yscale=NULL, xlim=NULL,
            ...)
## S3 method for class 'SpliceSites'
plot(x, col.typeI = "orange",
            col.typeI.window = "yellow",
            col.typeII = "red",
            add=FALSE, ylim=NULL, ...)
```

## Arguments

| | |
|---|---|
| `x` | object of class `Probes` or `SpliceSites`. |
| `col` | color argument for the probes. |
| `col.typeI` | color argument for the type I splice sites |
| `col.typeI.window` | color argument for the type I 'window' |
| `col.typeII` | color argument for the type II splice sites |
| `add` | add the plot to an existing plot. Make a new plot if 'FALSE' |
| `probepos.yscale` | scaling argument |
| `xlim, ylim` | range of plotting window |
| `xlab, ylab` | labels for the axis |
| `...` | optional parameters to be passed to the function `plot`. |

## Details

If the parameter `main` is not specified, the function tries to extract the attribute 'name' from `x`.

The two functions can be combined to display both objects on the same plot.

## Value

The range for the y-axis is returned whenever needed (see `invisible`).

## Author(s)

Laurent

## References

"Standardized output for putative alternative splicing; a R package as an application to combine them with microarray data", Gautier L. Dao C. and Yang U.C., 2003, submitted.

## See Also

SpliceSites-class

## Examples

```
data(spsites)

plot(spsites, main=attr(spsites, "name"))

sp.pData <- spsites@spsiteIpos.pData

##col <- as.integer(factor(sp.pData$tissue))

##plot(spsites, col.typeI=col, main=attr("name", spsites))
```

---

# plot.SpliceSitesGenomic

*Function to plot SpliceSitesGenomic objects*

---

## Description

Function to plot SpliceSitesGenomic objects.

## Usage

```
## S3 method for class 'SpliceSitesGenomic'
plot(x, col.variant = par("col"), col.exon = "white",
                        split = FALSE, main = names(x@variants), ...)
```

### Arguments

| | |
|---|---|
| x | SpliceSitesGenomic-class |
| col.variant | a vector of colors for the different variants. The colors are recycled as necessary. |
| col.exon | a vector of colors for the exons. The colors are recycled as necessary. |
| split | split the plot of the variants in individual plots |
| main | character to use as a title. Recycled as necessary. |
| ... | optional graphical parameters |

### Value

This function is used for its side-effect.

### Author(s)

Laurent

### See Also

SpliceSitesGenomic-class

### Examples

```
## a 10 bp window
seq.length <- as.integer(10)
## positions of the exons
spsiteIpos <- matrix(c(1, 3.5, 5, 9, 3, 4, 8, 10), nc=2)
## known variants
variants <- list(a=c(1,2,3,4), b=c(1,2,3), c=c(1,3,4))
##
n.exons <- nrow(spsiteIpos)

spvar <- new("SpliceSitesGenomic", spsiteIpos=spsiteIpos,
         variants=variants, seq.length=seq.length)

par(mfrow = c(3,1), mar = c(3.1, 2.1, 2.1, 1.1))

plot(spvar, split=TRUE, col.exon=rainbow(n.exons))
```

---

## Probes-class *Class "Probes"*

---

### Description

Information about a set of probes

### Objects from the Class

Objects can be created by calls of the form new("Probes", pos) or new("Probes", pos, info). The object are primarily storing the location of the probe on a matching sequence sequence. Optional information can be stored in the slot info (a data.frame).

### Slots

pos: Object of class "matrix". It expects one row per probe. The first column should give the start position while the second column should give the end position

info: Object of class "data.frame". Optional information one wishes to carry around can be stored here.

### Methods

**initialize** signature(.Object = "Probes"): ...

**show** signature(object): show minimal information

**plot** signature(x = "Probes", y = "missing"): plot the position of the probes. (see plot.Probes)

**plot** signature(x = "Probes", y = "SpliceSites"): plot the positions of the probes and the positions of the splice sites

### See Also

SpliceSites-class, SpliceExprSet-class,

### Examples

```
data(probes)

plot(probes)
```

---

## sort.SpliceExprSet *A function to sort a SpliceExprSet*

---

### Description

Sort the probes in a SpliceExprSet (and reflect this in all the relevant places.

### Usage

```
## S3 method for class 'SpliceExprSet'
sort(x, decreasing, fun = function(x) order(x@probes@pos[, 1]), reverse = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | a SpliceExprSet. |
| decreasing | currently ignored |
| fun | a function to do the sorting |
| reverse | return the reverse of the sorting order |
| ... | currently ignored |

### Value

An object of class SpliceExprSet

### Author(s)

Laurent

### See Also

SpliceExprSet-class

### Examples

```
data(spliceset)

s.spliceset <- sort.SpliceExprSet(spliceset)
```

---

## SpliceExprSet-class *Class "SpliceExprSet"*

---

### Description

A class to store probe expression values with alternative splicing information

### Objects from the Class

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

### Slots

`spliceSites`: Object of class `"SpliceSites"`. The probes and splice site information.

`probes`: Object of class `"Probes"`. The matching expression values.

`eset`: Object of class `"ExpressionSet"`. The matching expression values.

### Methods

**grid.plot** `signature(x = "SpliceExprSet", y = "missing")`: ...

**plot** `signature(x = "SpliceExprSet", y = "missing")`: a ploting method.

**show** `signature(object = "SpliceExprSet")`: a printing method.

**spliceSites** `signature(object = "SpliceExprSet")`: accessor.

### Author(s)

laurent@cbs.dtu.dk

### References

a manuscript in preparation

### See Also

as.data.frame.SpliceExprSet, sort.SpliceExprSet and SpliceSites-class

## Examples

```
data(eset, package="splicegear")
data(probes, package="splicegear")
data(spsites, package="splicegear")

spliceset <- new("SpliceExprSet", eset=eset,
                        probes=probes, spliceSites=spsites)

plot(spliceset)
```

---

`spliceset` *Example data for splicegear*

---

## Description

The putative splice variants for a reference sequence, the matching probes from the Affymetrix chip 'HG-U95A' and probe intensities from the 'dilution' dataset.

## Usage

```
#data(eset, package="splicegear")
#data(probes, package="splicegear")
#data(spsites, package="splicegear")
#data(spliceset, package="splicegear")
```

## Format

The formats are objects of class `ExpressionSet-class`, `Probes-class`, `SpliceSites-class` and `SpliceExprSet-class` respectively.

## Details

The attribute "name" is set to the ID of the Unigene cluster from which the reference sequence is taken.

## References

"PALSdb", ref. GeneLogic's dilution dataset.

## Examples

```
data(spliceset, package="splicegear")

plot(spliceset, main=attr(spliceset, "name"))
```

SpliceSites-class *Class "SpliceSites"*

## Description

A class to store (putative) splice sites

## Objects from the Class

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

## Slots

**`seq`:** Object of class `"character"`. The reference sequence.

**`seq.length`:** Object of class `"integer"`. The length for the reference sequence (used when the slot `seq` is set to `""`).

**`spsiteIpos`:** Object of class `"matrix"`. A two-columns matrix to store the begin and end positions of type I splice variant.

**`spsiteIIpos`:** Object of class `"integer"`. A vector to store the positions for type II splice variants.

**`spsiteIIIpos`:** Object of class `"matrix"`. Idem `spsiteIpos`, but for type III splice variants.

**`spsiteIpos.pData`:** Object of class `AnnotatedDataFrame`. Used to store covariate information related to the splice variants.

**`spsiteIIpos.pData`:** Object of class `AnnotatedDataFrame`.

**`spsiteIIIpos.pData`:** Object of class `AnnotatedDataFrame`.

## Methods

**show** `signature(object = "SpliceSites")`: A printing method.

**plot** `signature(x = "SpliceSites", y = "missing")`: A plotting method

## Author(s)

laurent@cbs.dtu.dk

## References

"Plenty of splicin' or 'can regular Affymetrix chips be used to observe alternative splicing ?'", Gautier L. et al., 2003, manuscript in preparation (and the title might have to chang...).

## See Also

`isSpliceSiteOnProbe`, `isProbeOnSpliceSite`, `plot.SpliceSites`, `spliceset`.

## Examples

```
data(spliceset)

print(spliceset)

par(mfrow=c(1,2))

plot(spliceset, main=attr(spliceset, "name"))

## filter out supporting matches with unique positions
filter.typeI <- function(x) {unique(x[duplicated(x), , drop=FALSE])}
spliceset.filter <- spliceset
sSites <- spliceset.filter@spliceSites
sSites@spsiteIpos <- filter.typeI(sSites@spsiteIpos)
spliceset.filter@spliceSites <- sSites
## plot the resulting new object
plot(spliceset.filter)
```

---

# SpliceSitesGenomic-class

*Class "SpliceSitesGenomic"*

---

## Description

A class to store alternative splicing information on a genomic point of view.

## Objects from the Class

Objects can be created by calls of the form `new("SpliceSitesGenomic", seq, seq.length, spsiteIpos, spsiteII`

## Slots

`variants`**:** Object of class `"list"`. There is one element per splice variant. Each element in the list should be a vector of integers. Each integer refers to an exon. The sequence of integers determines the sequence of exons in the splice variant.

`seq`**:** Object of class `"character"`, from class `"SpliceSites"`.

`seq.length`**:** Object of class `"integer"`, from class `"SpliceSites"`.

`spsiteIpos`**:** Object of class `"matrix"`, from class `"SpliceSites"`.

`spsiteIIpos`**:** Object of class `"integer"`, from class `"SpliceSites"`. This should not have any practical use in this class.

`spsiteIIIpos`**:** Object of class `"matrix"`, from class `"SpliceSites"`. This should not have any practical use in this class.

`spsiteIpos.pData`**:** Object of class `"AnnotatedDataFrame"`, from class `"SpliceSites"`.

`spsiteIIpos.pData`**:** Object of class `"AnnotatedDataFrame"`,from class `"SpliceSites"`. This should not have any practical use in this class.

`spsiteIIIpos.pData`**:** Object of class `"AnnotatedDataFrame"`,from class `"SpliceSites"`. This should not have any practical use in this class.

### Extends

Class "SpliceSites", directly.

### Methods

**plot** signature(x = "SpliceSitesGenomic", y = "missing"): a plotting method for demonstration purposes.

### See Also

SpliceSites-class and plot.SpliceSitesGenomic.

### Examples

```
## a 10 bp window
seq.length <- as.integer(10)
## positions of the exons
spsiteIpos <- matrix(c(1, 3.5, 5, 9, 3, 4, 8, 10), nc=2)
## known variants
variants <- list(a=c(1,2,3,4), b=c(1,2,3), c=c(1,3,4))

spvar <- new("SpliceSitesGenomic", spsiteIpos=spsiteIpos,
         variants=variants, seq.length=seq.length)

plot(spvar)
```

---

## split.SpliceSites *split an instance of SpliceSites*

---

### Description

Split an instance of SpliceSites into several instances of SpliceSites

### Usage

```
## S3 method for class 'SpliceSites'
split(x, f = list(typeI = NA, typeII = NA), drop=NULL, ...)
```

### Arguments

| | |
|---|---|
| x | an instance of class SpliceSites-class. |
| f | a list of two factors (see details). |
| drop | not used (here to keep R CMD check happy) |
| ... | see drop above. |

### Details

The split usually is performed on a factor. Two factors are required because of the two categories of splicing events (type I (deletion of a fragment of the reference sequence) and type II (insertion of an element of the reference sequence). A character can be used instead of a factor. In this case the covariates with the given name, in the slots spsiteIpos.pData and spsiteIIpos.pData respectively, are used to make the split. When equal to NA, the covariate named site will be used.

## Value

A list of objects of class `SpliceSites`

## See Also

`SpliceSites-class`

## Examples

```
data(spsites)
split(spsites)
```

# Index