Package 'RBPSpliceMap'

June 18, 2015

Type Package

Title Summarize CLIPSeq data

Version 1.0

Date 2015-06-18

Author Laure Le Calvez

Maintainer Laure Le Calvez <laure.le-calvez@laposte.net>

Depends R (>= 2.10.0), GenomicRanges (>= 1.17.20), GenomicAlignments (>= 1.1.16)

Description This package is intended to manipulate CLIPSeq data and summarize them along genomic regions. Typically, it can be used to draw RNA splice map such as those found in licatalosi et al. or using other summarize functions. It allows to select a chromosome, strand and interval of interest. It provides numerical and graphical output about the coverage of splice sites and their surroundings.

License GPL(>=2.0)

R topics documented:

RBPSpliceMap-package *Summarize CLIPSeq data*

Description

This package is intended to manipulate CLIPSeq data and summarize them along genomic regions. Typically it can be use to draw RNA splice map such as those found in licatalosi et al. or using other summarize functions. It allows to select a chromosome, strand and interval of interest. It provides numerical and graphical output about the coverage of splice sites and their surroundings.

Details

Package:	RBPSpliceMap
Type:	Package
Version:	1.0
Date:	2015-06-18
License:	GPL(>=2.0)

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

Examples

```
regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Normalized mean of coverages of ranges of interest on the 3 splice site
spMapList3SS = spMap(exonInterest, mapReadsGRanges, spSite = "3SS", padding =  c(40, 200), goal = "normMean
#Mean of coverages of ranges of interest on the 5 splice site
spMapList5SS = spMap(exonInterest, mapReadsGRanges, spSite = "5SS", padding =  c(100, 50), goal = "mean")
#3 splice site representation
plotSpMap(spMapList3SS)
#5 splice site representation
plotSpMap(spMapList5SS)
```

bamToGRanges *bamToGRanges*

Description

This function uses the readGAlignments function to read a bamfile and creates a GRanges object.

Usage

```
bamToGRanges(bamFile)
```

Arguments

bamFile	bamFile is the path of a bam file

Details

The returned GRanges object contains "seqnames", IRanges(start, end) and "strand". It is possible to have access to the different values vector with :
- seqnames(<GRangesName>)
- start(ranges(<GRangesName>))
- end(ranges(<GRangesName>))
- strand(<GRangesName>)

Value

GRanges object

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

readGAlignments from package GenomicAlignments
as

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file.
bamToGRanges(iCLIPData)
```

---

chromosomeSelect *chromosomeSelect*

Description

This function subsets a GRanges object by the requested chromosome.

Usage

```
chromosomeSelect(objectGRanges, chromosome)
```

Arguments

objectGRanges	GRanges object from which we want to subset by chromosome.
chromosome	chromosome of interest as a character (ex: "7" or "chr7") /!\ It should conform to the GRanges object notation.

Value

GRanges object

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

strandSelect
intervalSelect
mapReadsTreatment

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Select the chromosome of interest
chromosomeSelect(mapReadsGRanges, chromosome = "chr7")

regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Select the chromosome of interest
chromosomeSelect(mapReadsGRanges, chromosome = "chr7")
```

`coverageChr` *coverageChr*

Description

This function, from a GRanges object, computes the coverage (number of reads/nucleotide) for the chromosome of interest.

Usage

```
coverageChr(mapReadsGRanges, chromosome)
```

Arguments

`mapReadsGRanges`	GRanges object from where we want to select the part concerning the chromosome of interest.
`chromosome`	chromosome of interest as a character (ex: "7" or "chr7") /!\ It should conform to the GRanges object notation.

Details

The returned Rle (package S4Vectors) represents the coverage (number of reads/nucleotide) of the analysed chromosome.
No strand can be selected here (see `strandSelect`)

Value

Rle (package S4Vectors

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

`coverageVector`
`coverage`
`mapReadsTreatment`

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Calculate the coverage on a specified chromosome
coverageChr(mapReadsGRanges, chromosome = "chr7")
```

---

coverageVector *coverageVector*

Description

This function transforms a coverage Rle (package S4Vector) into a coverage vector on a specified region of the Rle.

Usage

```
coverageVector(cover, from = 0, to)
```

Arguments

`cover`	Rle (package S4Vector) with the coverage (typically `coverageChr`)
`from`	from = coordinate from where you want the coverage DEFAULT = 0
`to`	The copordinate of the last read you want on the coverage.

Details

The returned vector contains the number of reads at each position (nucleotide).

Value

Vector

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net

See Also

coverageChr
coverage
mapReadsTreatment

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Calculate the coverage on a specified chromosome
cover = coverageChr(mapReadsGRanges, chromosome = "chr7")
#Transforme coverage Rle into a coverage vector
coverageVector(cover, from = 810695, to = 810796)
```

---

exonInterestTreatment *exonInterestTreatment*

Description

Given a GRanges object corresponding to exon coordinates, this function creates a new GRanges object with modified coordinates depending on the splice site and the padding chosen for the analysis.

Usage

```
exonInterestTreatment(exonInterest, spSite, pE, pI)
```

Arguments

exonInterest	GRanges object containing coordinates of interest (thess coordinates match with exon ones)
spSite	Splice Site of interest ("5SS" or "3SS")
pE	padding exon. Number of nucleotides required in the exon
pI	padding intron. Number of nucleotides required in the intron

Details

Only strand = "+" or "-" are treated. If the input GRanges contains unspecified strands = "*", they are excluded.
The returned GRanges contains the genomic coordinates of the region to be analysed around the chosen splice site.
If an exon is at the beggining of the chromosome and, with the padding, coordinates become negatives, there will be a problem.

Value

GRanges object

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

Examples

```
regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Treatment of the GRanges object with exons of Interest
exonInterestTreatment(exonInterest, spSite = "3SS", pE = 40, pI = 200)
```

---

intervalSelect *intervalSelect*

Description

This function returns the part of GRanges object in the input ranges.

Usage

```
intervalSelect(objectGRanges, from, to)
```

Arguments

objectGRanges	GRanges object from where we want to select the part concerning the interval of interest.
from	Coordinate from where you want the coverage.
to	Coordinate where you want the coverage to end.

Value

GRanges object

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

strandSelect
chromosomeSelect
mapReadsTreatment

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Select the interval of interest
intervalSelect(mapReadsGRanges, from = 134621217, to = 134681857)
```

`mapReadsTreatment` *mapReadsTreatment*

Description

This function, from a GRanges object, computes the coverage (number of reads/nucleotide) for a region defined by "chromosome", "strand" and a range of interest ("from" and "to") for a GRanges object.

Usage

```
mapReadsTreatment(mapReadsGRanges, chromosome, strand, from, to)
```

Arguments

`mapReadsGRanges`	GRanges object containing mapped reads (generally from a bam file) (see `bamToGRanges`)
`chromosome`	chromosome of interest as a string (ex: "7" or "chr7") /!\ It should conform to the GRanges object notation
`strand`	strand of interest as a string ("+", "-", "*") "*" correspond to both strands
`from`	Coordinate from where you want the coverage.
`to`	Coordinate where you want the coverage to end.

Details

The returned coverage vector indicates the number of reads per nucleotide along the region of interest.

Value

Coverage Vector

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

`strandSelect`
`intervalSelect`
`coverageChr`
`coverageVector`

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Treatment of the GRanges object with mapped reads
mapReadsTreatment(mapReadsGRanges, chromosome = "chr7", strand = "+", from = 810695, to = 810796)
```

---

plotSpMap *plotSpMap*

Description

This function take as input the output of (spMap) and plot a RNA splice map around the chosen splice site.

Usage

```
plotSpMap(spMapList, ylim = c(0, max(spMapList[[1]])), type = "s", color = "black")
```

Arguments

spMapList	List of 4 vectors spMapList[[1]] : coverage vector spMapList[[2]] : splice site ("3SS" or "5SS") spMapList[[3]] : paddingExon (number) spMapList[[4]] : paddingIntron (number) (typically comming from spMap)
ylim	numeric vector of length 2, giving the y coordinates ranges.
type	what type of plot should be drawn.
color	default color of all points and lines

Details

padding : - paddingExon and paddingIntron correspond to the number of nucleotides to be analysed around the splice site in the exon and the intron
- This function creates a representation of the coverage of a "splice site" (x=0) between paddingIntron (x= pI-1) and paddingExon (x=pE)
- paddingIntron is used with -1 because x=0 is the splice site and part of the Intron
- The returned graph represents the coverage of the region of interest

Value

Graph

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

spMap
coverageVector

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Normalized mean of coverages of ranges of interest on the 3 splice site
spMapList3SS = spMap(exonInterest, mapReadsGRanges, spSite = "3SS", padding =  c(40, 200), goal = "normMean
#Mean of coverages of ranges of interest on the 5 splice site
spMapList5SS = spMap(exonInterest, mapReadsGRanges, spSite = "5SS", padding =  c(100, 50), goal = "mean")
#3 splice site representation
plotSpMap(spMapList3SS)
#5 splice site representation
plotSpMap(spMapList5SS)
```

---

spMap *spMap*

Description

This function returns a summarized coverage calculated from all coverage vectors provided in the GRanges object "exonInterest".

Usage

```
spMap(exonInterest, mapReadsGRanges, spSite, padding, goal, window = 50)
```

Arguments

exonInterest	GRanges object containing coordinates of interest (typically exon coordinates)
mapReadsGRanges	GRanges object containing mapped reads (generally from a bam file) (see bamToGRanges) (typically from a CLIPSeq experiment)
spSite	Splice Site of interest 5' or 3' splice site("5SS" or "3SS")
padding	A vector with the padding exon (pE) and the padding intron (pI) in the form of c(pE,pI) (pE and pI correspond to the number of nucleotides required in the exon and in the intron)
goal	Function, applied between the different coverage vectors to get the wanted result into the return coverage vector. This function can be "licatalosi" to get a normalized complexity, "normMean" to get a normalized mean or a basic R function. - "licatalosi" : normalized complexity (see Details) - "normMean" : normalized mean - r-base function (ex: "sum", "mean")
window	Only necessary with licatalosi function. Size chosen for the window. DEFAULT = 50

Details

The coverage vector represents the number of reads (depending on the chosen function "goal") at each nucleotide.
This function returns a list containing the calculated coverage vector, splice site, padding exon (pE) and padding intron (pI). You can access differents elements with : <listName>[[i]] (i = 1, 2, 3 or 4 respectively)
"licatalosi" : The function is based on the "normalized complexity map of PTB-RNA" described by Licatalosi et al. (HITS-CLIP yields genome-wide insights into brain alternative RNA processing). It takes as input a matrix of reads coverage where each row correspond to a different genomic region. Each row is normalized by dividing by the sum of the read coverage for the row.The gene region is splitted in N windows of length ("window") and for each window, the sum of the normalized coverage is calculated to obtain one normalized coverage vector representing the matrix. In licatalosi et al. complexity is defined as "the number of different transcripts with a CLIP tag in the window". The complexity vector correspond to the number of transcripts containing reads in each window. The normalized complexity is obtained by multiplying the complexity vector by the normalized coverage vector. The result, a normalized complexity coverage vector is returned by the function spMap.

Value

list(coverage vector, splice site, pE, pI)

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

exonInterestTreatment
mapReadsTreatment

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Normalized mean of coverages of ranges of interest on the 3 splice site
spMapList3SS = spMap(exonInterest, mapReadsGRanges, spSite = "3SS", padding =  c(40, 200), goal = "normMean
#Mean of coverages of ranges of interest on the 5 splice site
spMapList5SS = spMap(exonInterest, mapReadsGRanges, spSite = "5SS", padding =  c(100, 50), goal = "mean")
```

---

strandSelect *strandSelect*

Description

This function returns the part of the GRanges object for the input "strand".

Usage

```
strandSelect(objectGRanges, strand)
```

Arguments

objectGRanges	GRanges object from where we want to select the part concerning the strand of interest.
strand	strand of interest as a string ("+", "-", "*") "*" correspond to both strands and returns the entire GRanges object

Value

GRanges object

Author(s)

Laure Le Calvez <laure.le-calvez@laposte.net>

See Also

intervalSelect
chromosomeSelect
mapReadsTreatment

Examples

```
iCLIPData = system.file("extdata", "iCLIPData.bam", package="RBPSpliceMap", mustWork=TRUE)
#Create a GRanges object from a bam file
mapReadsGRanges = bamToGRanges(iCLIPData)
#Select the strand of interest
strandSelect(mapReadsGRanges, strand = "+")

regulatedExons = system.file("extdata", "regulatedExons.bed", package="RBPSpliceMap", mustWork=TRUE)
#Create GRanges object from a bedFile with exons of interest
data <- read.table(regulatedExons, header = TRUE)
colnames(data) = c(chr, start, end, ProbeID, pval, strand)
exonInterest = with(data, GRanges(chr, IRanges(start, end), strand))
#Select the strand of interest
strandSelect(regulatedExons, strand = "+")
```

Index