

# Package ‘SPLINTER’

April 4, 2025

**Type** Package

**Title** Splice Interpreter of Transcripts

**Version** 1.33.0

**Date** 2024-04-24

**URL** <https://github.com/dianalow/SPLINTER/>

**BugReports** <https://github.com/dianalow/SPLINTER/issues>

**Description** Provides tools to analyze alternative splicing sites, interpret outcomes based on sequence information, select and design primers for site validation and give visual representation of the event to guide downstream experiments.

**License** GPL-2

**LazyData** TRUE

**Depends** R (>= 3.6.0), grDevices, stats

**Imports** graphics, ggplot2, seqLogo, Biostrings, palign, biomaRt, GenomicAlignments, GenomicRanges, GenomicFeatures, Gviz, IRanges, S4Vectors, GenomeInfoDb, utils, plyr, stringr, methods, BSgenome.Mmusculus.UCSC.mm9, googleVis

**biocViews** ImmunoOncology, GeneExpression, RNASeq, Visualization, AlternativeSplicing

**Collate** primerpcr.R main\_splinter.R

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**Suggests** txdbmaker, BiocStyle, knitr, rmarkdown

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

**git\_branch** devel

**git\_last\_commit** 15c4210

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2025-04-03

**Author** Diana Low [aut, cre]

**Maintainer** Diana Low <lowdiana@gmail.com>

## Contents

acceptor.m . . . . .	3
addEnsemblAnnotation . . . . .	3
annotateEvents . . . . .	4
callPrimer3 . . . . .	5
checkPrimer . . . . .	6
compatible_cds . . . . .	7
compatible_tx . . . . .	7
donor.m . . . . .	7
eventOutcomeCompare . . . . .	8
eventOutcomeTranslate . . . . .	9
eventPlot . . . . .	10
extendROI . . . . .	11
extractSpliceEvents . . . . .	12
extractSpliceSites . . . . .	13
findCompatibleEvents . . . . .	14
findCompatibleExon . . . . .	15
findExactOverlaps . . . . .	16
findTermination . . . . .	16
findTX . . . . .	17
getPCRsizes . . . . .	18
getRegionDNA . . . . .	19
insertRegion . . . . .	19
makeROI . . . . .	20
makeUniqueIDs . . . . .	21
matchExons . . . . .	22
metaremove . . . . .	22
pcr_result1 . . . . .	23
plot_seqlogo . . . . .	23
primers . . . . .	24
psiPlot . . . . .	25
region_minus_exon . . . . .	26
removeRegion . . . . .	26
remvalue . . . . .	27
roi . . . . .	27
shapiroAcceptor . . . . .	28
shapiroDensity . . . . .	29
shapiroDonor . . . . .	29
splice_data . . . . .	30
splice_fasta . . . . .	31
splitPCRhit . . . . .	31
the cds . . . . .	32

<i>acceptor.m</i>	3
theexons . . . . .	32
valid_cds . . . . .	33
valid_tx . . . . .	33
<b>Index</b>	<b>34</b>

---

<code>acceptor.m</code>	<i>acceptor.m</i>
-------------------------	-------------------

---

### Description

Acceptor site mammalian frequency matrices for GT-AG pairs from SpliceDB

### Usage

```
data("acceptor.m")
```

### Format

The format is: num [1:4, 1:15] 9 31.03 12.5 42.36 8.44 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:4] "A" "C" "G" "T" ..\$ : chr [1:15] "V1" "V2" "V3" "V4" ...

### Source

<http://www.softberry.com/spldb/SpliceDB.html>

### References

Burset M., Seledtsov I., Solovyev V. (Nucl.Acids Res.,2000,28,4364-4375; Nucl. Acids Res.,2001,29,255-259)

### Examples

```
data(acceptor.m)
```

---

<code>addEnsemblAnnotation</code>	<i>addEnsemblAnnotation</i>
-----------------------------------	-----------------------------

---

### Description

Adds annotation to `extractSpliceEvents` object (if not present)

### Usage

```
addEnsemblAnnotation(data, species = "hsapiens")
```

**Arguments**

data `extractSpliceEvents` object  
species character. biomaRt species passed to retrieve annotation. Common species include: 'hsapiens', 'mmusculus'

**Value**

`extractSpliceEvents` object with annotated genes under \$geneSymbol

**Author(s)**

Diana Low

**See Also**

[http://asia.ensembl.org/info/data/biomart/biomart\\_r\\_package.html#biomartexamples](http://asia.ensembl.org/info/data/biomart/biomart_r_package.html#biomartexamples)

**Examples**

```
data_path<-system.file("extdata",package="SPLINTER")
splice_data<-extractSpliceEvents(data=paste(data_path,"/skipped_exons.txt",sep=""))
#splice_data<-addEnsemblAnnotation(data=splice_data,species="mmusculus")
```

---

annotateEvents

*annotateEvents*

---

**Description**

Gives detailed description of splicing event in terms of splicing outcome post translation. Currently supports exon skipping and intron retention events.

**Usage**

```
annotateEvents(  
  thedata,  
  db,  
  bsgenome,  
  outputdir,  
  full_output = FALSE,  
  output_prefix = "results"  
)
```

**Arguments**

thedata	list. output of extractSpliceEvents.
db	TxDb object
bsgenome	BSGenome object
outputdir	character. relative output directory to current location.
full_output	logical. writes out detailed text report and generate figures.
output_prefix	character. text prefix for full_output files.

**Value**

list containing information on (1) data.frame with splicing regions (2) splice event type

**Author(s)**

Diana LOW

---

callPrimer3	<i>callPrimer3</i>
-------------	--------------------

---

**Description**

call primer3 for a given set of DNAstringSet object

**Usage**

```
callPrimer3(
  seq,
  size_range = "150-500",
  Tm = c(57, 59, 62),
  name = "Primer1",
  primer3 = "primer3-2.3.7/bin/primer3_core",
  thermo.param = "primer3-2.3.7/src/primer3_config/",
  sequence_target = NULL,
  settings = "primer3-2.3.7/primer3web_v4_0_0_default_settings.txt"
)
```

**Arguments**

seq	DNAstring object, one DNA string for the given amplicon
size_range	default: '151-500'
Tm	melting temperature parameters default:c(55,57,58)
name	name of the amplicon in chr_start_end format
primer3	primer3 path
thermo.param	thermodynamic parameters folder

sequence\_target      If one or more targets is specified then a legal primer pair must flank at least one of them.

settings              text file for parameters

### Details

modified to include SEQUENCE\_TARGET as an option

### Value

data.frame of designed primers and parameters

### Author(s)

Altuna Akalin's modified Arnaud Krebs' original function further modified here by Diana Low

### Examples

```
### NOT RUN ###
# primer_results<-callPrimer3(seq='')
```

---

checkPrimer	<i>checkPrimer</i>
-------------	--------------------

---

### Description

checkPrimer

### Usage

```
checkPrimer(pp, genome, roi = NULL)
```

### Arguments

pp                    data.frame defining primers, or output of [callPrimer3](#). minimal columns = PRIMER\_LEFT\_SEQUENCE,PRIMER\_RIGHT\_SEQUENCE

genome              BSgenome object

roi                   [makeROI](#) object

### Value

list of GRanges with primer locations

### Author(s)

Diana Low

**Examples**

```
# create a primer pair
roi
primer_pair <- data.frame(PRIMER_LEFT_SEQUENCE="agctcttgaattggagctgac",
                          PRIMER_RIGHT_SEQUENCE="cttagaaagaacaggaaatcc",
                          stringsAsFactors=FALSE)
```

---

compatible_cds	<i>compatible_cds</i>
----------------	-----------------------

---

**Description**

compatible\_cds

**Examples**

```
data(compatible_cds)
## maybe str(compatible_cds) ; plot(compatible_cds) ...
```

---

compatible_tx	<i>compatible_tx</i>
---------------	----------------------

---

**Description**

compatible\_tx

**Examples**

```
data(compatible_tx)
## maybe str(compatible_tx) ; plot(compatible_tx) ...
```

---

donor.m	<i>donor.m</i>
---------	----------------

---

**Description**

Donor site mammalian frequency matrices for GT-AG pairs from SpliceDB

**Usage**

```
data("donor.m")
```

**Format**

The format is: num [1:4, 1:9] 34.1 36.2 18.3 11.4 60.4 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:4] "A" "C" "G" "T" ..\$ : chr [1:9] "V1" "V2" "V3" "V4" ...

**Source**

<http://www.softberry.com/spldb/SpliceDB.html>

**References**

Burset M., Seledtsov I., Solovyev V. (Nucl.Acids Res.,2000,28,4364-4375; Nucl. Acids Res.,2001,29,255-259)

**Examples**

```
data(donor.m)
```

---

```
eventOutcomeCompare  eventOutcomeCompare
```

---

**Description**

Compares two sequences and gives differences if there's a switch from 1->2 if seq2 is NULL, assume seq1 is a list of length 2 to compare

**Usage**

```
eventOutcomeCompare(
  seq1,
  seq2 = NULL,
  genome,
  direction = TRUE,
  fullseq = TRUE,
  verbose = FALSE
)
```

**Arguments**

seq1	GRangesList
seq2	GRangesList
genome	BSGenome object
direction	logical. Report direction of sequence change.
fullseq	logical. Report full sequences.
verbose	logical. turn messages on/off.



**Value**

list containing  
 (1) tt : PairwiseAlignmentsSingleSubject pairwise alignment  
 (2) eventtypes : string detailing primary event classification

**Author(s)**

Diana LOW

**Examples**

```
suppressMessages(library(BSgenome.Mmusculus.UCSC.mm9))
bsgenome<-BSgenome.Mmusculus.UCSC.mm9
eventOutcomeCompare(seq1=compatible_cds$hits[[1]],seq2=region_minus_exon,
  genome=bsgenome,direction=TRUE)
```

---

eventOutcomeTranslate *eventOutcomeTranslate*

---

**Description**

translates sequences, reports if NMD or NTC

**Usage**

```
eventOutcomeTranslate(
  seq1,
  genome,
  direction = FALSE,
  fullseq = TRUE,
  verbose = FALSE
)
```

**Arguments**

seq1	GRangesList
genome	BSGenome object
direction	logical. Report direction of sequence change.
fullseq	logical. Output full AA sequence.
verbose	logical. turn messages on/off.

**Value**

list of translated sequences

**Author(s)**

Diana LOW

**Examples**

```
suppressMessages(library(BSgenome.Mmusculus.UCSC.mm9))
bsgenome<-BSgenome.Mmusculus.UCSC.mm9
translation_results<-eventOutcomeTranslate(compatible_cds,genome=bsgenome,
direction=TRUE)
```

---

eventPlot	<i>eventPlot</i>
-----------	------------------

---

**Description**

eventPlot

**Usage**

```
eventPlot(
  transcripts,
  roi_plot = NULL,
  bams = c(),
  names = c(),
  annoLabel = c("Gene A"),
  rspan = 1000,
  pfam_dom = NULL,
  showAll = TRUE
)
```

**Arguments**

transcripts	GRanges object
roi_plot	GRanges object region to plot
bams	character vector of bam file locations
names	character vector of name labels
annoLabel	character. annotation label
rspan	integer or NULL. number of basepairs to span from roi. if NULL, will consider whole gene of roi
pfam_dom	optional GRanges object of PFAM domains from UCSC Tables.
showAll	logical. TRUE = display splice junctions of entire view or FALSE = just roi.

**Value**

a Gviz plot of genomic region

**Author(s)**

Diana Low

**Examples**

```
# define BAM files
data_path<-system.file("extdata", package="SPLINTER")
mt<-paste(data_path, "/mt_chr14.bam", sep="")
wt<-paste(data_path, "/wt_chr14.bam", sep="")

# plot results
eventPlot(transcripts=valid_tx, roi_plot=roi, bams=c(wt, mt),
          names=c('wt', 'mt'), rspan=1000)
```

---

`extendROI`*extendROI*

---

**Description**

extend the span of the current ROI by n number of up/downstream exon(s) by modifying `roi_range` within the `makeROI` object while retaining legacy sites by keeping `$roi` and `$flank`

**Usage**

```
extendROI(roi, tx, up = 0, down = 0, type = 1)
```

**Arguments**

<code>roi</code>	<code>makeROI</code> object
<code>tx</code>	<code>GRangesList</code> transcript list to pull regions from
<code>up</code>	integer. number of exons to extend upstream
<code>down</code>	integer. number of exons to extend downstream
<code>type</code>	integer. 1=full cassette, 2=flank only

**Value**

`makeROI` object with modified ranges

**Examples**

```
extendROI(roi, valid_tx, up=1)
```

---

`extractSpliceEvents`    *extractSpliceEvents*

---

### Description

Extracts the location of target, upstream and downstream splice sites Used for calculations and genome visualizations

### Usage

```
extractSpliceEvents(
  data = NULL,
  filetype = "mats",
  splicetype = "SE",
  fdr = 1,
  inclusion = 1,
  start0 = TRUE
)
```

### Arguments

<code>data</code>	character. path to file
<code>filetype</code>	character. type of splicing output. c('mats','custom'). see Details.
<code>splicetype</code>	character. c('SE', 'RI', 'MXE', 'A5SS', 'A3SS')
<code>fdr</code>	numeric. false discovery rate filter range [0,1]
<code>inclusion</code>	numeric. splicing inclusion range, takes absolute value
<code>start0</code>	boolean 0-base start

### Details

filetype 'custom' should provide a 9-column tab-delimited text file with the following columns: ID (Ensembl gene id), Symbol (gene name), chr, strand, exonStart, exonEnd, exon2Start, exon2End, upstreamStart, upstreamEnd, downstreamStart, downstreamEnd eg. ENSG0000012345 chr1 + 3 4 5 6 1 2 7 8

### Value

list containing information on

- (1) original file type
- (2) splice event type
- (3) data.frame with splicing regions

### Author(s)

Diana Low

**See Also**

[http://rnaseq-mats.sourceforge.net/user\\_guide.htm](http://rnaseq-mats.sourceforge.net/user_guide.htm) for MATS file definition

**Examples**

```
data_path<-system.file("extdata",package="SPLINTER")
splice_data<-extractSpliceEvents(data=paste(data_path,"/skipped_exons.txt",sep=""))
```

---

extractSpliceSites      *extractSpliceSites*

---

**Description**

Extracts and formats to bed the location of target, upstream and downstream splice sites

**Usage**

```
extractSpliceSites(  
  df,  
  target = "SE",  
  site = "donor",  
  motif_range = c(-3, 6),  
  start0 = TRUE  
)
```

**Arguments**

df	extractSpliceEvents object
target	the target site to extract. See Details.
site	character donor or acceptor
motif_range	numeric vector of splice position to extract
start0	boolean 0-base start

**Details**

target : the site to extract the sequence from. It can be either the event in question (SE, RI, MXE - first exon, MXE2 - second exon, A5SSlong, A5SSshort, A3SSlong, A3SSshort, upstream or downstream). If this function is used in conjunction with [shapiroDonor](#) or [shapiroAcceptor](#) to compute scores, then most likely it will be run twice - once for the event, and the other either up- or downstream as a comparison.

**Value**

GRanges object

**Author(s)**

Diana Low

**See Also**[http://rnaseq-mats.sourceforge.net/user\\_guide.htm](http://rnaseq-mats.sourceforge.net/user_guide.htm) for MATS file definition**Examples**

```
data_path<-system.file("extdata",package="SPLINTER")
splice_data<-extractSpliceEvents(data=paste(data_path,"/skipped_exons.txt",sep=""))
splice_sites<-extractSpliceSites(splice_data,target="SE")
```

---

findCompatibleEvents    *findCompatibleEvents*

---

**Description**

Which transcript contains the event? Each event has 2 possibilities, as long as the transcript fulfills one, it passes the test Has to be exact (inner junctions)

**Usage**

```
findCompatibleEvents(tx, tx2 = NULL, roi, sequential = TRUE, verbose = FALSE)
```

**Arguments**

tx	GRangesList object of transcripts
tx2	optional GRangesList object of transcripts if tx is list of cds
roi	<a href="#">makeROI</a> object containing event information
sequential	logical. Exons have to appear sequentially to be considered compatible
verbose	logical. printouts and messages.

**Details**

Separates into event/region1 and 2 for the alternative case

**Value**

list of length 4  
 (1) GRangesList  
 (2) Hits status [c]=coding; [nc]=non-coding  
 (3) ct - compatible transcripts  
 (4) tt - total transcripts

**Author(s)**

Diana Low

**Examples**

```
compatible_cds <- findCompatibleEvents(valid_cds,roi=roi,verbose=TRUE)
```

---

findCompatibleExon     *findCompatibleExon*

---

**Description**

Finds compatible exon in annotation with the one present in roi object

**Usage**

```
findCompatibleExon(tx, roi, verbose = FALSE)
```

**Arguments**

tx	GRangesList object of transcripts
roi	<a href="#">makeROI</a> object containing event information
verbose	logical. printouts and messages.

**Value**

list of length 3  
(1) GRangesList hits  
(2) Number of transcripts  
(3) Original number of input transcripts

**Author(s)**

Diana Low

**Examples**

```
compatible_exons <- findCompatibleExon(valid_cds,roi)
```

---

`findExactOverlaps`      *findExactOverlaps*

---

**Description**

Internal function similar to `findSpliceOverlaps` but only preserves internal flanks

**Usage**

```
findExactOverlaps(query, subject, sequential = FALSE, verbose = FALSE)
```

**Arguments**

<code>query</code>	GRanges object
<code>subject</code>	GRanges object
<code>sequential</code>	logical. TRUE if exons are sequential.
<code>verbose</code>	logical. report intermediate output

**Value**

Hits object

**Author(s)**

Diana Low

---

`findTermination`      *findTermination*

---

**Description**

Internal function to find the first stop codon that occurs in the AA sequence, returns their position and the resulting truncated protein

**Usage**

```
findTermination(s1)
```

**Arguments**

<code>s1</code>	character. protein sequence
-----------------	-----------------------------



**Value**

list containing  
(1) stop1 : stop position  
(2) s1 : sequence truncated to first stop

**Author(s)**

Diana LOW

---

findTX	<i>findTX</i>
--------	---------------

---

**Description**

Given an ENSEMBL id, find all transcripts that matches id

**Usage**

```
findTX(id, db, tx, valid = FALSE, verbose = FALSE)
```

**Arguments**

id	character. transcript identification (currently ENSEMBL gene names)
db	TxDB object
tx	GRangesList
valid	logical. check if in multiples of 3 [TRUE] for CDS translation.
verbose	logical. turn messages on/off.

**Value**

GRangesList

**Author(s)**

Diana Low

**Examples**

```
valid_cds <- findTX(id=splice_data$data[2,]$ID, tx=thecds, db=txdb, valid=FALSE)
```

---

`getPCRsizes`*getPCRsizes*

---

**Description**

returns length of product given a GRanges span and GRangesList of transcripts

**Usage**

```
getPCRsizes(pcr_span, txlist, verbose = FALSE)
```

**Arguments**

<code>pcr_span</code>	GRanges object
<code>txlist</code>	GRangesList object
<code>verbose</code>	logical. report intermediate output.

**Value**

data.frame of transcript names with detected sizes in basepairs

**Author(s)**

Diana Low

**Examples**

```
suppressMessages(library(BSgenome.Mmusculus.UCSC.mm9))
bsgenome<-BSgenome.Mmusculus.UCSC.mm9
## create a primer pair
## for actual use, obtain primer pair from primer design (callPrimer3)
primer_pair <- data.frame(PRIMER_LEFT_SEQUENCE="agctcttgaatggagctgac",
                          PRIMER_RIGHT_SEQUENCE="cttagaaagaacaggaatcc",
                          stringsAsFactors=FALSE)

## confirm location
cp<-checkPrimer(primer_pair,bsgenome,roi)
cp

## get the PCR sizes
pcr_result1 <- getPCRsizes(cp,theexons)
```

---

getRegionDNA	<i>getRegionDNA</i>
--------------	---------------------

---

**Description**

get DNA sequence give a region of interest

**Usage**

```
getRegionDNA(roi, genome, introns = FALSE)
```

**Arguments**

roi	makeROI object
genome	BSgenome object
introns	TRUE/FALSE. whether to include intronic (lowercase) DNA. By default returns only exonic (uppercase) DNA.

**Value**

list of  
(1) DNA sequence (2) Junction start (for primer design)

**Author(s)**

Diana Low

**Examples**

```
suppressMessages(library(BSgenome.Mmusculus.UCSC.mm9))
bsgenome<-BSgenome.Mmusculus.UCSC.mm9
getRegionDNA(roi,bsgenome)
```

---

insertRegion	<i>insertRegion</i>
--------------	---------------------

---

**Description**

inserts a region (exon or intron) into roi object

**Usage**

```
insertRegion(subject, roi)
```

**Arguments**

subject            GRangesList  
 roi                [makeROI](#) object containing region of interest (to insert). refer to [makeROI\(\)](#).

**Details**

in the case of intron retention, replaces exon with intron retention range `reduce()` the GRanges in question

**Value**

GRanges object

**Author(s)**

Diana Low

**Examples**

```
#Inserts the exon defined in roi GRanges object from a GRanges/GRangesList
region_minus_exon
region_with_exon<-insertRegion(region_minus_exon,roi)
```

---

makeROI

*makeROI*

---

**Description**

Creates an object to store information about the splice site (region of interest) including flanking regions and alternative splice outcome

**Usage**

```
makeROI(df, type = "SE")
```

**Arguments**

df                 data.frame object from [extractSpliceEvents](#)  
 type              type of splicing event c("SE","RI","MXE","A5SS","A3SS")

**Value**

a list containing

- (1) type : splice type
- (2) name : ID of transcript
- (3) roi : GRanges object of splice site
- (4) flank : GRanges object of flanking exons of splice site
- (5) roi\_range : GRangesList of splice site and its alternative outcome based on type

**Author(s)**

Diana Low

**Examples**

```
single_record<-splice_data$data[which(grepl("Prmt5",splice_data$data$Symbol)),]  
roi <- makeROI(single_record,type="SE")
```

---

makeUniqueIDs	<i>makeUniqueIDs</i>
---------------	----------------------

---

**Description**

Makes unique ID names from event location

**Usage**

```
makeUniqueIDs(ddata)
```

**Arguments**

ddata            extractSpliceEvents object

**Value**

original extractSpliceEvents list object with unique ID appended to data accessor

**Author(s)**

Diana Low

**Examples**

```
data_with_id<-makeUniqueIDs(splice_data)
```

matchExons

*matchExons*

---

**Description**

Internal function to help match the inner coordinates of a 2/3 cassette checks if reference and subject matches

**Usage**

```
matchExons(ref, subject)
```

**Arguments**

ref                   GRanges object

subject               GRanges object

**Value**

logical. check if exons match (TRUE) or not (FALSE)

**Author(s)**

Diana Low

---

metaremove

*metaremove*

---

**Description**

helper function to remove metadata from GRanges object

**Usage**

```
metaremove(x)
```

**Arguments**

x                     GRanges or GRangesList

**Value**

GRanges or GRangesList

---

`pcr_result1`*pcr\_result1*

---

**Description**`pcr_result1`**Examples**`data(pcr_result1)`

---

`plot_seqlogo`*plotting sequence logo*

---

**Description**

Plots the sequence logo of a given set of FASTA sequences

**Usage**`plot_seqlogo(fasta_seq)`**Arguments**`fasta_seq` DNASTringSet or path to fasta-formatted file**Value**

sequence logo image

**Author(s)**

Diana Low

**Examples**

```
head(splice_fasta)
plot_seqlogo(Biostrings::DNASTringSet(splice_fasta$V2))
```

---

 primers

*primers*


---

### Description

primers designed using Primer3 for sample data

### Usage

```
data("primers")
```

### Format

A data frame with 5 observations on the following 28 variables.

*i* a numeric vector  
 PRIMER\_LEFT\_SEQUENCE a character vector  
 PRIMER\_RIGHT\_SEQUENCE a character vector  
 PRIMER\_LEFT\_TM a numeric vector  
 PRIMER\_RIGHT\_TM a numeric vector  
 PRIMER\_LEFT\_pos a numeric vector  
 PRIMER\_LEFT\_len a numeric vector  
 PRIMER\_RIGHT\_pos a numeric vector  
 PRIMER\_RIGHT\_len a numeric vector  
 PRIMER\_PAIR\_PENALTY a numeric vector  
 PRIMER\_LEFT\_PENALTY a numeric vector  
 PRIMER\_RIGHT\_PENALTY a numeric vector  
 PRIMER\_LEFT\_GC\_PERCENT a numeric vector  
 PRIMER\_RIGHT\_GC\_PERCENT a numeric vector  
 PRIMER\_LEFT\_SELF\_ANY\_TH a numeric vector  
 PRIMER\_RIGHT\_SELF\_ANY\_TH a numeric vector  
 PRIMER\_LEFT\_SELF\_END\_TH a numeric vector  
 PRIMER\_RIGHT\_SELF\_END\_TH a numeric vector  
 PRIMER\_LEFT\_HAIRPIN\_TH a numeric vector  
 PRIMER\_RIGHT\_HAIRPIN\_TH a numeric vector  
 PRIMER\_LEFT\_END\_STABILITY a numeric vector  
 PRIMER\_RIGHT\_END\_STABILITY a numeric vector  
 PRIMER\_LEFT\_TEMPLATE\_MISPRIMING a numeric vector  
 PRIMER\_RIGHT\_TEMPLATE\_MISPRIMING a numeric vector  
 PRIMER\_PAIR\_COMPL\_ANY\_TH a numeric vector  
 PRIMER\_PAIR\_COMPL\_END\_TH a numeric vector  
 PRIMER\_PAIR\_PRODUCT\_SIZE a numeric vector  
 PRIMER\_PAIR\_TEMPLATE\_MISPRIMING a numeric vector



**Value**

Dataframe of primer design results

**Examples**

```
data(primers)
```

---

psiPlot

*psiPlot*

---

**Description**

Plots percentage spliced in (PSI) values in terms of inclusion levels

**Usage**

```
psiPlot(df = NULL, type = "MATS", sample_labels = c("Sample 1", "Sample 2"))
```

**Arguments**

df	data.frame containing PSI values
type	character. either 'MATS' output (will read in MATS headers) or 'generic' (provide 4 or 6 column data.frame)
sample_labels	x-axis labels for the plot

**Value**

bar plot of PSI values

**Author(s)**

Diana Low

**Examples**

```
#we give inclusion and skipped numbers as reads
#this will be converted into percentages
df<-data.frame(inclusion1=c("6,4,6"),skipped1=c("10,12,12"),inclusion2=c("15,15,15"),
               skipped2=c("3,3,4"),stringsAsFactors = FALSE)
psiPlot(df,type='generic')
```

---

region_minus_exon	<i>region_minus_exon</i>
-------------------	--------------------------

---

**Description**

region\_minus\_exon

**Examples**

```
data(region_minus_exon)
## maybe str(region_minus_exon) ; plot(region_minus_exon) ...
```

---

removeRegion	<i>removeRegion</i>
--------------	---------------------

---

**Description**

removes a region (exon) from a GRanges or GRangesList

**Usage**

```
removeRegion(subject, roi)
```

**Arguments**

subject	GRanges or GrangesList object
roi	<a href="#">makeROI</a> object containing GRanges range (to remove)

**Value**

GRanges object

**Author(s)**

Diana Low

```
# Removes the exon defined in roi GRanges object from a GRanges/GRangesList compatible_cds$hits[[1]]
region_minus_exon<-removeRegion(compatible_cds$hits[[1]],roi)
```

---

remvalue	<i>remvalue</i>
----------	-----------------

---

**Description**

helper function to remove metadata from GRanges object used within metaremove

**Usage**

```
remvalue(x)
```

**Arguments**

x GRanges or GRangesList

**Value**

GRanges or GRangesList

---

roi	<i>roi</i>
-----	------------

---

**Description**

roi

**Usage**

```
data("roi")
```

**Value**

List containing region of interest information

**Examples**

```
data(roi)
```

---

shapiroAcceptor	<i>shapiroAcceptor</i>
-----------------	------------------------

---

### Description

Shapiro's score of acceptor site (range is from -13 [intron] to +1 [exon]) is:  $100 * ((t1 - l1)/(h1 - l1) + (t2 - l2)/(h2 - l2))/2$ , where t1 is the sum of the best 8 of 10 percentages at positions -13 to -4, l1 is the sum of the lowest 8 of 10 percentages at position -13 to -4, h1 is the sum of the highest 8 of 10 percentages at positions -13 to -4, t2 is the sum of percentages at positions -3 to +1, l2 is the sum of the lowest percentages at positions -3 to +1, and h2 is the sum of the highest percentages at positions -3 to +1

### Usage

```
shapiroAcceptor(reference_fasta, target_fasta)
```

### Arguments

reference\_fasta      vector of strings or DNAStringSet of reference splice list

target\_fasta        vector of strings or DNAStringSet of fasta to score

### Value

data.frame with Shapiro scores

### Author(s)

Diana Low

### See Also

<http://www.softberry.com/spldb/SpliceDB.html>

### Examples

```
library(BSgenome.Mmusculus.UCSC.mm9)
bsgenome <- BSgenome.Mmusculus.UCSC.mm9
data_path<-system.file("extdata",package="SPLINTER")
splice_data<-extractSpliceEvents(data=paste(data_path,"/skipped_exons.txt",sep=""))
splice_sites<-extractSpliceSites(splice_data,site="acceptor")
acceptor.ss<-getSeq(bsgenome,splice_sites)
##sacceptor<-shapiroAcceptor(acceptor.m,acceptor.ss)
```

---

shapiroDensity	<i>shapiroDensity</i>
----------------	-----------------------

---

**Description**

convenience function for plotting Shapiro score density

**Usage**

```
shapiroDensity(ctrl_scores, treat_scores, sample = c(1, 2))
```

**Arguments**

ctrl_scores	output of shapiroDonor or shapiroAcceptor
treat_scores	output of shapiroDonor or shapiroAcceptor
sample	samplenames

**Value**

density plot of Shapiro scores

**Author(s)**

Diana Low

---

shapiroDonor	<i>shapiroDonor</i>
--------------	---------------------

---

**Description**

Shapiro and Senapathy (1987) have developed a method to score the strength of a splice site based on percentages of each nucleotide at each position. Shapiro's score of donor site (range is from -3 [exon] to +7 [intron]) is :  $100 * (t - \min) / (\max - \min)$ , where t is the sum of percentages at positions -3 to +7, min is the sum of the lowest percentages at positions -3 to +7, and max is the sum of the highest percentages at positions -3 to +7.

**Usage**

```
shapiroDonor(reference_fasta, target_fasta)
```

**Arguments**

reference_fasta	vector of strings or DNASTringSet of reference splice list
target_fasta	vector of strings or DNASTringSet of fasta to score

**Value**

data.frame with Shapiro scores

**Author(s)**

Diana Low

**See Also**

<http://www.softberry.com/spldb/SpliceDB.html>

**Examples**

```
library(BSgenome.Mmusculus.UCSC.mm9)
bsgenome <- BSgenome.Mmusculus.UCSC.mm9
data_path<-system.file("extdata",package="SPLINTER")
splice_data<-extractSpliceEvents(data=paste(data_path,"/skipped_exons.txt",sep=""))
splice_sites<-extractSpliceSites(splice_data)
donor.ss<-getSeq(bsgenome,splice_sites)
##sdonor<-shapiroDonor(donor.m,donor.ss)
```

---

splice\_data

*splice\_data*

---

**Description**

splice\_data

**Usage**

```
data("splice_data")
```

**Value**

List containing splice event file information

**Examples**

```
data(splice_data)
```

---

splice_fasta	<i>splice_fasta</i>
--------------	---------------------

---

**Description**

splice\_fasta

**Usage**

```
data("splice_fasta")
```

**Format**

A data frame with 0 observations on the following 2 variables.

V1 a numeric vector

V2 a numeric vector

**Value**

Dataframe of region and fasta sequence

**Examples**

```
data(splice_fasta)
```

---

splitPCRhit	<i>splitPCRhit</i>
-------------	--------------------

---

**Description**

splits the PCR alignment into the two AS conditions

**Usage**

```
splitPCRhit(res, hitlist)
```

**Arguments**

res result from [getPCRsizes](#)

hitlist [findCompatibleEvents](#) object

**Value**

list of 2 data.frame objects with isoform name (ID) and length of PCR product (bp) matching Type 1 or Type 2 transcripts

**Author(s)**

Diana Low

**Examples**

```
## as getPCRsizes gives you all PCR bands when the primers are used,
## splitPCRhit will determine which bands are relevant to the target
relevant_pcr_bands<-splitPCRhit(pcr_result1,compatible_tx)
```

---

the cds	<i>the cds</i>
---------	----------------

---

**Description**

the cds

**Usage**

```
data("the cds")
```

**Value**

List containing GRanges info

**Examples**

```
data(the cds)
```

---

the exons	<i>the exons</i>
-----------	------------------

---

**Description**

the exons

**Usage**

```
data("the cds")
```

**Value**

List containing GRanges info

**Examples**

```
data(the exons)
```



---

valid_cds	<i>valid_cds</i>
-----------	------------------

---

**Description**

valid\_cds

**Usage**

```
data("valid_cds")
```

**Value**

GRangesList

**Examples**

```
data(valid_cds)
```

---

valid_tx	<i>valid_tx</i>
----------	-----------------

---

**Description**

valid\_tx

**Value**

GRangesList

**Examples**

```
data(valid_tx)  
## maybe str(valid_tx) ; plot(valid_tx) ...
```

# Index

## \* datasets

- acceptor.m, 3
- compatible\_cds, 7
- compatible\_tx, 7
- donor.m, 7
- pcr\_result1, 23
- primers, 24
- region\_minus\_exon, 26
- roi, 27
- splice\_data, 30
- splice\_fasta, 31
- thecds, 32
- theexons, 32
- valid\_cds, 33
- valid\_tx, 33

## \* internal

- findExactOverlaps, 16
- findTermination, 16
- matchExons, 22
- metaremove, 22
- remvalue, 27

acceptor.m, 3

addEnsemblAnnotation, 3

annotateEvents, 4

callPrimer3, 5, 6

checkPrimer, 6

compatible\_cds, 7

compatible\_tx, 7

donor.m, 7

eventOutcomeCompare, 8

eventOutcomeTranslate, 9

eventPlot, 10

extendROI, 11

extractSpliceEvents, 3, 4, 12, 20

extractSpliceSites, 13

findCompatibleEvents, 14, 31

findCompatibleExon, 15

findExactOverlaps, 16

findTermination, 16

findTX, 17

getPCRsizes, 18, 31

getRegionDNA, 19

insertRegion, 19

makeROI, 6, 11, 14, 15, 19, 20, 20, 26

makeUniqueIDs, 21

matchExons, 22

metaremove, 22

pcr\_result1, 23

plot\_seqlogo, 23

primers, 24

psiPlot, 25

region\_minus\_exon, 26

removeRegion, 26

remvalue, 27

roi, 27

shapiroAcceptor, 13, 28

shapiroDensity, 29

shapiroDonor, 13, 29

splice\_data, 30

splice\_fasta, 31

splitPCRhit, 31

thecds, 32

theexons, 32

valid\_cds, 33

valid\_tx, 33