Package 'Guitar'

October 17, 2017

Type Package

Title Guitar

Version 1.14.0

Date 2016-7-14

Author Jia Meng

Maintainer Jia Meng <jia.meng@xjtlu.edu.cn>

Description The package is designed for visualization
of RNA-related genomic features with respect to the landmarks of RNA transcripts, i.e., transcription starting site, start codon, stop codon and transcription ending site.

License GPL-2

Depends Rsamtools, GenomicFeatures, rtracklayer, GenomicAlignments, GenomicRanges, ggplot2, grid, IRanges

biocViews Sequencing, SplicedAlignment, Alignment, DataImport, RNASeq, MethylSeq, QualityControl, Transcription, Coverage

NeedsCompilation no

R topics documented:

Guitar-package *Guitar*

Description

RNA Landmarks Guided Transcriptomic View of RNA-related Genomic Features.

Details

The package is designed for transcriptomic visualization of RNA-related genomic features represented with genome-based coordinates with respect to the landmarks of RNA transcripts, i.e., transcription starting site, start codon, stop codon and transcription ending site.

Author(s)

Jia Meng <jia.meng@hotmail.com>

References

~~ Literature or other references for background information ~~

Examples

```
# read genomic features
narrowPeak <- system.file(
  "extdata", "m6A_hg19_1000peaks_macs2.narrowPeak",
  package="Guitar")

# bam imported as GAlignments
m6A_Bcell <- narrowPeaktoGRanges(narrowPeak)

# generate a list of genomic features
m6A_Bcell_1 <- m6A_Bcell[1:300]
m6A_Bcell_2 <- m6A_Bcell[301:600]
m6A_Bcell_3 <- m6A_Bcell[601:900]
feature_hg19 <- list(m6A_Bcell_1, m6A_Bcell_2, m6A_Bcell_3)
names(feature_hg19) <- c("m6A_1","m6A_2","m6A_3")

# Make Guitar coordiantes
txdb_file <- system.file("extdata", "hg19_toy.sqlite",
                         package="Guitar")
txdb <- loadDb(txdb_file)
gc_txdb <- makeGuitarCoordsFromTxDb(txdb,noBins =10)


# Plot
GuitarPlot(feature_hg19,
           GuitarCoordsFromTxDb = gc_txdb)
```

BED12toGRangesList *BED12toGRangesList*

Description

read bed12 format into R as GRangesList object

Usage

```
BED12toGRangesList(filepath, header)
```

Arguments

filepath	the path where the bed12 file is located
header	whether the bed12 file has header or not, default: FALSE

Details

The function read bed12 into R as GRangesList object, with the introns spliced out, making it different from other functions such as import.bed.

Value

The returned GRangsList object has the same number of GRangesLists as the number of rows in the bed12 file.

Author(s)

Jia Meng <jia.meng@hotmail.com>

References

For more infomration about the bed format, please refer to: https://genome.ucsc.edu/FAQ/FAQformat.html#format1

Examples

```
bed12=system.file("extdata", "m6A_mm10_exomePeak_1000peaks_bed12.bed", package="Guitar")
m6A_HepG2 <- BED12toGRangesList(bed12)
```

combinedGuitarPlot *combinedGuitarPlot*

Description

combine multiple GuitarPlots, especially when they are from different species and cannot directly handled with GuitarPlot function together.

Usage

```
combinedGuitarPlot(ct, comLength = c(0.136,0.459, 0.405))
```

Arguments

ct	Count of overlapping features, which is can be the output from GuitarPlot function. See example.
comLength	a vector of 3 elements,indicating the length of 5'UTR, CDS and 3'UTR in the figure. Default: c(0.136,0.459, 0.405), which is counted from the default setting of hg19 UCSC gene annotation.

Details

combine multiple GuitarPlots, especially when they are from different species.

Value

A figure showing the transcriptomic distribution of the genomic features will be generated. Post-editing with Adobe Illustrator or other graphic software is recommended.

Examples

```
# read genomic features
narrowPeak <- system.file(
  "extdata", "m6A_hg19_1000peaks_macs2.narrowPeak",
  package="Guitar")

# bam imported as GAlignments
m6A_Bcell <- narrowPeaktoGRanges(narrowPeak)

# generate a list of genomic features
m6A_Bcell_1 <- m6A_Bcell[1:300]
m6A_Bcell_2 <- m6A_Bcell[301:600]
m6A_Bcell_3 <- m6A_Bcell[601:900]
m6A_Bcell_4 <- m6A_Bcell[201:900]

# Make Guitar coordiantes
txdb_file <- system.file("extdata", "hg19_toy.sqlite",
                         package="Guitar")
txdb <- loadDb(txdb_file)
gc_txdb <- makeGuitarCoordsFromTxDb(txdb,noBins =10)

# Guitar plot 1
```

```
feature1_hg19 <- list(m6A_Bcell_1, m6A_Bcell_2)
names(feature1_hg19) <- c("m6A_1","m6A_2")
ct1 <- GuitarPlot(feature1_hg19, returnCount = TRUE,
           GuitarCoordsFromTxDb = gc_txdb)

# Guitar plot 2
feature2_hg19 <- list(m6A_Bcell_3, m6A_Bcell_4)
names(feature2_hg19) <- c("m6A_3","m6A_4")
ct2 <- GuitarPlot(feature2_hg19, returnCount = TRUE,
           GuitarCoordsFromTxDb = gc_txdb)

# combine two Gutiar Plot
ct <- rbind(ct1,ct2)
combinedGuitarPlot(ct)
```

`getNeighborhood` *getNeighborhood*

Description

This functon extract the neighborhood DNA regions of a GRangesList object. The neighborhood DNA regions represent the promoter region (5'end) and complementary DNA region on the 3'end, which we call "Tail" here. The 5'end neighborhood DNA can often be useful as a kind of negative control region that is not associated with a clear biological function.

This function is different from the "promoters" function available from GenomicRanges R package, which is designed to extract only the promoter region (5' side DNA), but not the 3' side DNA region.

Usage

```
getNeighborhood(comp, side = 5, Width = 1000)
```

Arguments

`comp`	a GRangesList object, whose 5' or 3' side DNA neighborhood will be extracted.
`side`	either 5 or 3. When it is 5, the promoter region of the GRangesList object will be extracted; if it is 3, the region on the 3' side complementary to the promoter region will be extracted.
`Width`	The width of the neighborhood region will be extracted. default: 1000

Details

All inner list elements of the input GRangesList object must have the same strand

Value

A GRangesList, of the same length as the input, will be returned. You may use unlist to convert it into a GRanges object.

Author(s)

Jia Meng <jia.meng@hotmail.com>

Examples

```
gr1 <-
  GRanges(seqnames = "chr2", ranges = IRanges(3000, 6000),
          strand = "+", score = 5L, GC = 0.45)
gr2 <-
  GRanges(seqnames = c("chr1", "chr1"),
          ranges = IRanges(c(7000,13000), width = 3),
          strand = c("+"), score = 3:4, GC = c(0.3, 0.5))
grl <- GRangesList("gr1" = gr1, "gr2" = gr2)
promoter <- getNeighborhood(grl,5)
promoter
promoter_GRanges <- unlist(promoter)
promoter_GRanges
```

GuitarPlot *GuitarPlot*

Description

Plot the transcriptomic distribution of genomic features

Usage

```
GuitarPlot(gfeatures, GuitarCoordsFromTxDb = NA,
  txdb = NA, genome = NA,
  noBins = 10, saveToPDFprefix = NA,
  returnCount = FALSE, includeNeighborDNA = FALSE,
  maximalFeatureAmbiguity=5, rescaleComponent=TRUE, fill=FALSE, adjust=1)
```

Arguments

gfeatures	genomic features, which can be GRanges, GRangesList, GAlignment.
GuitarCoordsFromTxDb	Guitar coordiantes generated from makeGuitarCoordsFromTxDb object
txdb	If GuitarCoordsFromTxDb is not provided, you may optionally provide a TranscriptDb object, and the Guitar coordiante will be generated accordingly from it.
genome	If neither GuitarCoordsFromTxDb nortxdb is not provided, you may optionally provide genome assembly number, gene annotation will be automatically downloaded from UCSC, and the Guitar coordiante will be generated accordingly from it.
noBins	The resolution of the analysis. Default: 100
saveToPDFprefix	Whether the figure generated should be saved into a PDF file.
returnCount	Whether the count information should be returned. This information is from countOverlaps between the provided gfeatures and Guitar coordiantes. It is not recommended to use this information.
includeNeighborDNA	Whether neighborhood DNA regions should be included in the GuitarPlot.

`maximalFeatureAmbiguity`
: Features that overlap with more than this number of transcripts will not be used in the analysis. If a genomic feature overlaps with multiple transcripts but the number doesn't exceed this threshold, its weight will be evenly divided. Default: 5

`rescaleComponent`
: Adjustment of the bandwidth. Default: 1.

`fill`
: Whether calculate the relative location-standardized density or not. If set "TRUE", the density of tracks will be standardized with respect to each location. Default: FALSE

`adjust`
: Whether calculate the relative location-standardized density or not. If set "TRUE", the density of tracks will be standardized with respect to each location. Default: 1

Details

This function plots the transcriptomic distribution of genomic features. It is designed for a fast usage of the Guitar package without the needs to go into the details.

Value

A figure showing the transcriptomic distribution of the genomic features will be generated. Post-editing with Adobe Illustrator or other graphic software is recommended.

Examples

```
# read genomic features
narrowPeak <- system.file(
  "extdata", "m6A_hg19_1000peaks_macs2.narrowPeak",
  package="Guitar")

# bam imported as GAlignments
m6A_Bcell <- narrowPeaktoGRanges(narrowPeak)

# generate a list of genomic features
m6A_Bcell_1 <- m6A_Bcell[1:300]
m6A_Bcell_2 <- m6A_Bcell[301:600]
m6A_Bcell_3 <- m6A_Bcell[601:900]
feature_hg19 <- list(m6A_Bcell_1, m6A_Bcell_2, m6A_Bcell_3)
names(feature_hg19) <- c("m6A_1","m6A_2","m6A_3")

# Make Guitar coordiantes
txdb_file <- system.file("extdata", "hg19_toy.sqlite",
                         package="Guitar")
txdb <- loadDb(txdb_file)
gc_txdb <- makeGuitarCoordsFromTxDb(txdb,noBins =10)


# Plot
GuitarPlot(feature_hg19,
           GuitarCoordsFromTxDb = gc_txdb)
```

makeGuitarCoordsFromGRangesList
makeGuitarCoordsFromGRangesList

Description

make Guitar Coordinates From a GRangesList object

Usage

```
makeGuitarCoordsFromGRangesList(comp, noBins = 100, collapseGene = FALSE, width = 51)
```

Arguments

comp	A GRangesList object
noBins	The number of sections the "transcript" of the GRangesList object will be equally divided into. This is the resolution of the Guitar coordinates. The larger noBins is, the clearer the visualization will be; however, more computation time and memory resource will be required. You may want to set a smaller number when using a slow computer. Default: 100.
collapseGene	Whether merge the Guitar coordinates from different transcripts. Default: FALSE.
width	The width of each check points. Default: 51

Value

Guitar coordinates will be generated, which provides a fast reference between genomic coordinates and the transcriptomic coordinates. By default, the returned value should be in GRanges format; if collapseGene = TRUE, GRangesList object will be returned.

Author(s)

Jia Meng <jia.meng@hotmail.com>

Examples

```
gr1 <-  GRanges(seqnames = "chr2", ranges = IRanges(3000, 6000),
          strand = "+", score = 5L, GC = 0.45)
gr2 <-  GRanges(seqnames = c("chr1", "chr1"),
          ranges = IRanges(c(7000,13000), width = 3),
          strand = c("+"), score = 3:4, GC = c(0.3, 0.5))
grl <- GRangesList("gr1" = gr1, "gr2" = gr2)
tc <- makeGuitarCoordsFromGRangesList(grl, noBins = 5)
tc
mcols(tc)
```

makeGuitarCoordsFromTxDb *makeGuitarCoordsFromTxDb*

Description

Make a Guitar Coordinates from TranscriptDb object, i.e., making Guitar coordinates for 8 different components, including, 5'UTR, CDS, 3'UTR, lncRNA, Pomoter and Tail of mRNA and lncRNA. Additional filters will discard transcripts that are too short or has too much ambigous on Genome to increase the sensitivity of the analysis.

Usage

```
makeGuitarCoordsFromTxDb(txdb, maximalAmbiguity = 3,
    minimalComponentLength = 100, minimalNcRNALength = 300,
    noBins = 100)
```

Arguments

`txdb`	A transcriptDb object, which can be generated from makeTxDbFromUCSC or other functions.
`maximalAmbiguity`	If a transcript overlap with more number of transcripts than this number, this transcript will be used in the analysis. By filtering out a number of transcripts, this filter also decrease memory usage and computation time. Default: 3.
`minimalComponentLength`	The mimimal length of the components (5'UTR, CDS, 3'UTR) of a mRNA. Unfortunately, some mRNAs do not all 3 components or some components can be too short and cannot provide effective resolution for the analysis. These mRNAs will be filtered out from the analysis. Default: 100
`minimalNcRNALength`	non-coding RNAs with length smaller than this value will not be used in the analysis.
`noBins`	The number of sections the "transcript" of the GRangesList object will be equally divided into. This is the resolution of the Guitar coordinates. The larger noBins is, the clearer the visualization will be; however, more computation time and memory resource will be required. You may want to set a smaller number when using a slow computer. Default: 100.

Value

A Guitar coordiantes (GRanges object) will be returned, with Transcript ID, the relative position of each GRanges on the RNA transcript, the interval (bp) between different coordiantes on a transcript component: Front (DNA), Back (DNA), 5'UTR, CDS, 3'UTR, lncRNA.

Author(s)

Jia Meng <jia.meng@hotmail.com>

Examples

```
txdb_file <- system.file("extdata", "hg19_knownGene_sample.sqlite",
                         package="GenomicFeatures")
txdb <- loadDb(txdb_file)
gc_txdb <- makeGuitarCoordsFromTxDb(txdb,noBins =3)
gc_txdb
```

`narrowPeaktoGRanges` *narrowPeaktoGRanges*

Description

read the narrowpeak format from MACS software into GRangesList object.

NAME_peaks.narrowPeak is BED6+4 format file which contains the peak locations together with peak summit, pvalue and qvalue. You can load it to UCSC genome browser. Definition of some specific columns are: 5th: integer score for display 7th: fold-change 8th: -log10pvalue 9th: -log10qvalue 10th: relative summit position to peak start.

Usage

```
narrowPeaktoGRanges(file)
```

Arguments

`file`	a string specifies where the narrow peak file is located.

Value

A GRanges object will be returned.

References

MACS2 software: https://github.com/taoliu/MACS/tree/master/MACS2

Examples

```
narrowPeak=system.file("extdata", "m6A_hg19_1000peaks_macs2.narrowPeak", package="Guitar")
m6A_Bcell <- narrowPeaktoGRanges(narrowPeak) # bam imported as GAlignments
m6A_Bcell
```

Index