Package ‘branchpointer’

October 17, 2017

Type Package

Title Prediction of intronic splicing branchpoints

Version 1.0.0

Date 2017-03-01

Author Beth Signal

Maintainer Beth Signal <b.signal@garvan.org.au>

Description Predicts branchpoint probability for sites in intronic branchpoint windows. Queries can be supplied as intronic regions; or to evaluate the effects of mutations, SNPs.

License BSD_3_clause + file LICENSE

LazyData TRUE

Depends caret, R(>= 3.4)

Imports plyr, kernlab, gbm, stringr, cowplot, ggplot2, biomaRt, Biostrings, parallel, utils, BSgenome.Hsapiens.UCSC.hg38, rtracklayer, GenomicRanges, IRanges, S4Vectors, data.table

Suggests knitr, BiocStyle

RoxygenNote 6.0.1

VignetteBuilder knitr

biocViews Software, GenomeAnnotation, GenomicVariation, MotifAnnotation

NeedsCompilation no

R topics documented:

`getQueryLoc` *Find the closest 3’ and 5’ exons to a branchpointer query*

Description

Finds the closest annotated exons from genomic co-ordinates in a branchpointer query GRanges

Usage

```
getQueryLoc(query, queryType, maxDist = 50, filter = TRUE, exons,
  useParallel = FALSE, cores = 1)
```

Arguments

`query` branchpointer query GenomicRanges must have chromosome at position 2, genomic co-ordinate at position 3, and strand at position 4.

`queryType` type of query file (`"SNP"` or `"region"`)

`maxDist` maximum distance a SNP can be from an annotated 3’ exon.

`filter` remove SNP queries prior to finding finding nearest exons.

`exons` data.frame containing exon co-ordinates. Should be produced by gtfToExons()

`useParallel` use parallelisation to speed up code? (reccomended for > 500 query entries)

`cores` number of cores to use in parallelisation (default = `1`)

Value

GenomicRanges with the query and its location relative to the 3’ and 5’ exons

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",package = "branchpointer")
exons <- gtfToExons(smallExons)

querySNP <- system.file("extdata","SNP_example.txt", package = "branchpointer")
query <- readQueryFile(querySNP,queryType = "SNP")
query <- getQueryLoc(query,queryType = "SNP",exons = exons, filter = FALSE)
```

`gtfToExons` *Convert GTF file to exon location file*

Description

Converts a GTF annotation to exon locations

Usage

```
gtfToExons(gtf)
```

Arguments

`gtf`	file containing the gtf annotation.

Value

exon annotation GRanges

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",
package = "branchpointer")
exons <- gtfToExons(smallExons)
```

`makeRegions` *Make branchpoint window regions*

Description

Genrate branchpoint window regions corresponding to annotated exon(s) within a queried gene, transcript or exon id

Usage

```
makeRegions(id, idType, exons)
```

Arguments

`id`	identifier for the query gene/transcript/exon id
`idType`	type of id to match in the exon annotation file (`"gene_id"`, `"transcript_id"`, or `"exon_id"`)
`exons`	data.frame containing exon co-ordinates. Should be produced by gtfToExons()

Value

Granges with formatted query

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",package = "branchpointer")
exons <- gtfToExons(smallExons)
windowquery <- makeRegions("ENSG00000139618", "gene_id", exons)
windowquery <- makeRegions("ENST00000357654", "transcript_id", exons)
windowquery <- makeRegions("ENSE00003518965", "exon_id", exons)
```

---

plotBranchpointWindow *Plots branchpointer predictions*

Description

Plots branchpointer predictions

Usage

```
plotBranchpointWindow(queryName, predictions, probabilityCutoff = 0.5,
  plotMutated = FALSE, plotStructure = TRUE, exons)
```

Arguments

queryName	query id used to identify the SNP or region
predictions	Granges object generated by predictBranchpoints()
probabilityCutoff	probability score cutoff value for displaying U2 binding energy
plotMutated	plot alternative sequence predicitons alongside reference sequence predictions
plotStructure	plot structures for gene and 3' exon containing and skipping isoforms
exons	Granges containing exon co-ordinates. Should be produced by gtfToExons()

Value

ggplot2 plot with branchpoint features in the specified intronic region

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",package = "branchpointer")
exons <- gtfToExons(smallExons)
genome <- BSgenome.Hsapiens.UCSC.hg38::BSgenome.Hsapiens.UCSC.hg38

querySNP <- system.file("extdata","SNP_example.txt", package = "branchpointer")
query <- readQueryFile(querySNP,queryType = "SNP")
query <- getQueryLoc(query,queryType = "SNP",exons = exons, filter = FALSE)
```

```
predictions <- predictBranchpoints(query,queryType = "SNP", BSgenome = genome)
plotBranchpointWindow(query$id[1], predictions,
plotMutated = TRUE, exons = exons)
```

plotStructure *Plots transcript structures*

Description

Plots transcript structures

Usage

```
plotStructure(exonID, exons, keepTranscripts = "overlapping")
```

Arguments

exonID	id of the exon to plot
exons	Granges containing exon co-ordinates.
keepTranscripts	which transcripts to plot ("overlapping" or "withExon") "overlapping" will plot all transcripts overlapping the exon, whereas "withExon" will plot all transcripts containing the exon.

Value

ggplot2 plot transcript structures

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",
package = "branchpointer")
exons <- gtfToExons(smallExons)
plotStructure(exonID = "ENSE00001184784.4", exons)
```

`predictBranchpoints` *Predict branchpoint probability scores*

Description

predicts branchpoint probability scores for each query site.

Usage

```
predictBranchpoints(query, uniqueId = "test", queryType,
  workingDirectory = ".", genome = NA, bedtoolsLocation = NA,
  BSgenome = NULL, useParallel = FALSE, cores = 1, rmChr = FALSE)
```

Arguments

`query` branchpointer query GenomicRanges

`uniqueId` unique string identifier for intermediate .bed and .fa files.

`queryType` type of branchpointer query. "SNP" or "region".

`workingDirectory` directory where intermediate .bed and .fa are located

`genome` .fa genome file location

`bedtoolsLocation` bedtools binary location (which bedtools)

`BSgenome` BSgenome object

`useParallel` use parallelisation to speed up code?

`cores` number of cores to use in parallelisation (default = 1)

`rmChr` remove "chr" before chromosome names before writing bed file. Required if genome sequence names do not contain "chr"

Value

GenomicRanges object with branchpoint probaility scores for each site in query

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",
package = "branchpointer")
exons <- gtfToExons(smallExons)
genome <- BSgenome.Hsapiens.UCSC.hg38::BSgenome.Hsapiens.UCSC.hg38

querySNP <- system.file("extdata","SNP_example.txt", package = "branchpointer")
query <- readQueryFile(querySNP,queryType = "SNP")
query <- getQueryLoc(query,queryType = "SNP",exons = exons, filter = FALSE)
predictions <- predictBranchpoints(query,queryType = "SNP",BSgenome = genome)
```

`predictionsToStats` *Convert SNP branchpoint predictions across the branchpoint window to an intronic summary*

Description

Takes predictions of branchpoint probabilities from a reference and alternative SNP and summarises the effect(s) of the SNP.

Usage

```
predictionsToStats(query, predictions, probabilityCutoff = 0.5,
  probabilityChange = 0.2)
```

Arguments

`query` query GRanges containing all SNP ids to be summarised

`predictions` site-wide branchpoint proability predictions produced from predictBranchpoints()

`probabilityCutoff`
Value to be used as the cutoff for discriminating branchpoint sites from non-branchpoint sites (default = `0.5`)

`probabilityChange`
Minimum probability score change required to call a branchpoint site as deleted or created by a SNP (default = `0.2`)

Value

GRanges with summarised branchpoint changes occuring within the intron due to a SNP.

Author(s)

Beth Signal

Examples

```
smallExons <- system.file("extdata","gencode.v24.annotation.small.gtf",
package = "branchpointer")
exons <- gtfToExons(smallExons)
genome <- BSgenome.Hsapiens.UCSC.hg38::BSgenome.Hsapiens.UCSC.hg38

querySNP <- system.file("extdata","SNP_example.txt", package = "branchpointer")
query <- readQueryFile(querySNP,queryType = "SNP")
query <- getQueryLoc(query,queryType = "SNP",exons = exons, filter = FALSE)
predictions <- predictBranchpoints(query,queryType = "SNP",BSgenome = genome)
snpStats <- predictionsToStats(query,predictions)
```

`readQueryFile` *Read a query file*

Description

Reads and formats a query file, generated from snpToQuery, or manually generated. Converts unstranded SNPs to two entries for each strand. Checks for duplicate names and replaces if found.

Usage

```
readQueryFile(queryFile, queryType)
```

Arguments

`queryFile`	tab delimited file containing query information. For intronic regions should be in the format: region id, chromosome name, region start, region id, strand. For SNP variants should be in the format: SNP id, chromosome name, SNP position, strand, reference allele (A/T/C/G), alternative allele (A/T/C/G)
`queryType`	type of query file (`"SNP"` or `"region"`)

Value

data.frame with formatted query

Author(s)

Beth Signal

Examples

```
querySNP <- system.file("extdata","SNP_example.txt", package = "branchpointer")
query <- readQueryFile(querySNP,queryType = "SNP")

queryIntron <- system.file("extdata","intron_example.txt", package = "branchpointer")
query <- readQueryFile(queryIntron,queryType = "region")
```

`snpToQuery` *Gets a branchpointer formatted query from refsnp ids*

Description

Searches Biomart for refsnp ids, and pulls genomic location and sequence identity information Reformats alleles so each query has only one alternative allele

Usage

```
snpToQuery(refSNP, mart.snp)
```

Arguments

`refSNP`	Vector of refsnp ids
`mart.snp`	biomaRt mart object specifying the BioMart database and dataset to be used

Value

formatted SNP query data.frame

Author(s)

Beth Signal

Examples

```
mart.snp <- biomaRt::useMart("ENSEMBL_MART_SNP", dataset="hsapiens_snp", host="www.ensembl.org")
query <- snpToQuery("rs17000647", mart.snp)
```

Index