Package ‘proActiv’

October 25, 2025

Title Estimate Promoter Activity from RNA-Seq data

Version 1.19.3

Description Most human genes have multiple promoters that control the expression of different isoforms. The use of these alternative promoters enables the regulation of isoform expression pre-transcriptionally. Alternative promoters have been found to be important in a wide number of cell types and diseases. proActiv is an R package that enables the analysis of promoters from RNA-seq data. proActiv uses aligned reads as input, and generates counts and normalized promoter activity estimates for each annotated promoter. In particular, proActiv accepts junction files from TopHat2 or STAR or BAM files as inputs. These estimates can then be used to identify which promoter is active, which promoter is inactive, and which promoters change their activity across conditions. proActiv also allows visualization of promoter activity across conditions.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

RoxygenNote 7.3.1

Imports AnnotationDbi, BiocParallel, data.table, dplyr, DESeq2, IRanges, GenomicRanges, GenomicFeatures, GenomicAlignments, GenomeInfoDb, ggplot2, gplots, graphics, methods, rlang, scales, S4Vectors, SummarizedExperiment, stats, tibble, txdbmaker

Suggests GenomeInfoDbData, testthat, rmarkdown, knitr, Rtsne, gridExtra

URL `https://github.com/GoekeLab/proActiv`

biocViews RNASeq, GeneExpression, Transcription, AlternativeSplicing, GeneRegulation, DifferentialSplicing, FunctionalGenomics, Epigenetics, Transcriptomics, Preprocessing

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/proActiv

git_branch devel

git_last_commit 042c9f7

git_last_commit_date 2025-07-31

Repository Bioconductor 3.22

Date/Publication 2025-10-24

Author Deniz Demircioglu [aut] (ORCID:
<https://orcid.org/0000-0001-7857-0407>),
Jonathan Göke [aut],
Joseph Lee [cre]

Maintainer Joseph Lee <joseph.lee@u.nus.edu>

Contents

`boxplotPromoters` *Visualizes promoter activity and gene expression with boxplots*

Description

Visualizes promoter activity and gene expression with boxplots

Usage

```
boxplotPromoters(
  result,
  geneId,
  geneName = NULL,
  filterInternal = TRUE,
  col = NULL
)
```

Arguments

`result`	A SummarizedExperiment object return by proActiv, with assays giving promoter counts and activity with gene expression stored as metadata. rowData contains promoter metadata and absolute promoter activity summarized across conditions. Condition must be provided.
`geneId`	A character vector. A single gene id. This identifier must correspond to the identifier in the promoter annotation.
`geneName`	A character vector. Common gene name to be displayed on plot. Optional. Defaults to NULL.
`filterInternal`	A boolean. Determines if internal promoters should be removed from the plot. Defaults to TRUE.
`col`	A character vector of colours to be used for plotting.

Value

A list of length 3. Each entry is a plot corresponding to absolute promoter activity, relative promoter activity and gene expression.

Examples

```
files <- list.files(system.file('extdata/vignette/junctions',
                       package = 'proActiv'),
                       full.names = TRUE, pattern = 'replicate5')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
result <- proActiv(files = files,
                       promoterAnnotation  = promoterAnnotation,
                       condition = rep(c('A549', 'HepG2'), each=1),
                       fileLabels = NULL,
                       ncores = 1)
plots <- boxplotPromoters(result, "ENSG00000076864.19")
```

`calculateJunctionReadCounts` *Calculate the total number of junction reads overlapping with the introns of each promoter for the input junction file*

Description

Calculate the total number of junction reads overlapping with the introns of each promoter for the input junction file

Usage

```
calculateJunctionReadCounts(
  promoterCoordinates,
  intronRanges,
  file = "",
  fileType = "",
  genome = ""
)
```

Arguments

`promoterCoordinates`	A GRanges object containing promoter coordinates and reduced exon coordinates by gene
`intronRanges`	A Granges object containing the annotated unique intron ranges. These ranges will be used for counting the reads
`file`	character path for the input junction bed or bam file
`fileType`	character type of the junction bed file. Either 'tophat', 'star' or 'bam'
`genome`	character genome version

Value

The total number of junction reads overlapping with each promoter for the input annotated intron ranges

---

`calculatePromoterReadCounts`

Calculate the promoter read counts using junction read counts approach for all the input junction files

Description

Calculate the promoter read counts using junction read counts approach for all the input junction files

Usage

```
calculatePromoterReadCounts(
  promoterAnnotation,
  files = NULL,
  fileLabels = NULL,
  fileType = NULL,
  genome = NULL,
  numberOfCores = 1
)
```

Arguments

`promoterAnnotation`	A PromoterAnnotation object containing the reduced exon ranges, annotated intron ranges, promoter coordinates and the promoter id mapping
`files`	A character vector. The list of junction or BAM files for which the junction read counts will be calculated
`fileLabels`	A character vector. The labels of junction or BAM files for which the junction read counts will be calculated. These labels will be used as column names for the output data.frame object
`fileType`	A character. Type of the junction bed or bam file, either 'tophat', 'star' or 'bam
`genome`	A character. Genome version used. Must be specified if input is a BAM file. Defaults to NULL

`numberOfCores`	A numeric value. The number of cores to be used for counting junction reads. Defaults to 1 (no parallelization). This parameter will be used as an argument to BiocParallel::bplapply

Value

A data.frame object. The number of junction reads per promoter (rows) for each sample (cols)

`getAbsolutePromoterActivity` *Prepare the absolute promoter activity table including the promoter and gene ids*

Description

Prepare the absolute promoter activity table including the promoter and gene ids

Usage

```
getAbsolutePromoterActivity(
  junctionReadCounts,
  promoterAnnotation,
  log2 = TRUE,
  pseudocount = 1
)
```

Arguments

`junctionReadCounts`	Matrix of junction read counts (rows: promoters, cols: samples)
`promoterAnnotation`	A PromoterAnnotation object containing the intron ranges, promoter coordinates and the promoter id mapping
`log2`	Logical indicating whether log2 read counts should be used (default: TRUE) or not
`pseudocount`	Number to be used for log2 as pseudocount if log2 is TRUE

Value

data.frame of absolute promoter activity with promoter and gene ids

---

`getAlternativePromoters`

Identifies alternative promoters.

Description

Identifies alternative promoters.

Usage

```
getAlternativePromoters(
  result,
  referenceCondition,
  minAbs = 0.25,
  minRel = 0.05,
  maxPval = 0.05,
  promoterFC = 2,
  geneFC = 1.5
)
```

Arguments

`result`	A SummarizedExperiment object with assays giving promoter counts, activity and gene expression (output from proActiv). rowData contains promoter metadata and absolute promoter activity summarized across conditions. Condition must be provided.
`referenceCondition`	A character vector. The reference condition to be compared. Samples corresponding to all other conditions will be compared to this samples in this current condition.
`minAbs`	A numeric value. Minimum value for promoter to be active in absolute terms. Defaults to 0.25.
`minRel`	A numeric value. Minimum value for promoter to be active in relative terms. Defaults to 0.05.
`maxPval`	A numeric value. Adjusted p-value threshold for detecting alternative promoters. Defaults to 0.05.
`promoterFC`	A numeric value. Minimum fold change for a promoter in the current condition compared to all other conditions. Promoters must have at least this magnitude of fold change for alternative usage.
`geneFC`	A numeric value. Maximum fold change for gene expression. To identify alternative promoter usage independent of changes in gene expression, limit the gene expression fold change.

Value

A list of length 2. Each entry is a dataframe summarizing up-regulated and down-regulated promoters and their corresponding genes, if any.

Examples

```
files <- list.files(system.file('extdata/vignette/junctions',
                       package = 'proActiv'),
                       full.names = TRUE, pattern = 'replicate5')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
result <- proActiv(files = files,
                       promoterAnnotation  = promoterAnnotation,
                       condition = rep(c('A549', 'HepG2'), each=1),
                       fileLabels = NULL,
                       ncores = 1)
alternativePromoters <- getAlternativePromoters(result, "A549")
```

`getGeneExpression` *Prepare the gene expression table including the gene ids*

Description

Prepare the gene expression table including the gene ids

Usage

```
getGeneExpression(absolutePromoterActivity)
```

Arguments

`absolutePromoterActivity`	data.frame of absolute promoter activity with promoter and gene ids

Value

data.frame of gene expression with gene ids#'

`getRelativePromoterActivity` *Prepare the relative promoter activity table including the promoter and gene ids*

Description

Prepare the relative promoter activity table including the promoter and gene ids

Usage

```
getRelativePromoterActivity(absolutePromoterActivity, geneExpression)
```

Arguments

`absolutePromoterActivity`	data.frame of absolute promoter activity with promoter and gene ids
`geneExpression`	data.frame of gene expression with gene ids

Value

data.frame of relative promoter activity with promoter and gene ids

---

`integrateProactiv` *Integrate multiple proActiv runs*

Description

Integrate multiple proActiv runs

Usage

```
integrateProactiv(res1, res2, ..., promoterAnnotation, renormalize = TRUE)
```

Arguments

`res1`	A summarizedExperiment object returned by proActiv
`res2`	A summarizedExperiment object returned by proActiv
`...`	Additional summarizedExperiment objects returned by proActiv
`promoterAnnotation`	Promoter annotation object used to create proActiv runs
`renormalize`	Whether to renormalize counts after merging. Defaults to TRUE

Value

A SummarizedExperiment object with assays giving promoter counts and activity with gene expression. rowData contains promoter metadata and absolute promoter activity summarized across conditions (if condition is provided)

Examples

```
f1 <- list.files(system.file('extdata/vignette/junctions',
                             package = 'proActiv'),
                 full.names = TRUE, pattern = 'A549')
f2 <- list.files(system.file('extdata/vignette/junctions',
                             package = 'proActiv'),
                 full.names = TRUE, pattern = 'HepG2')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
res1 <- proActiv(files = f1, promoterAnnotation  = promoterAnnotation,
                 condition = rep('A549',3))
res2 <- proActiv(files = f2, promoterAnnotation = promoterAnnotation,
                 condition = rep('HepG2',3))
res <- integrateProactiv(res1, res2, promoterAnnotation = promoterAnnotation)
```

`normalizePromoterReadCounts` *Normalize promoter read counts using DESeq2*

Description

Normalize promoter read counts using DESeq2

Usage

```
normalizePromoterReadCounts(promoterReadCounts)
```

Arguments

`promoterReadCounts`	A data.frame object. The number of junction reads per promoter (rows) for each sample (cols)

Value

A data.frame object. The normalized number of junction reads per promoter (rows) for each sample (cols) using DESeq2 counts function. Requires 'DESeq2' package to be installed

---

`plotHeatmap` *Visualizes heatmap of features for samples*

Description

Visualizes heatmap of features for samples

Usage

```
plotHeatmap(
  result,
  by = "absolutePromoterActivity",
  features = NULL,
  cex.legend = 0.75,
  cex.row = NULL,
  cex.col = NULL,
  row.margin = 5,
  col.margin = 12,
  col = NULL,
  breaks = NULL,
  palette = "bluered"
)
```

Arguments

`result`	A SummarizedExperiment object return by proActiv, with assays giving promoter counts and activity with gene expression stored as metadata. rowData contains promoter metadata and absolute promoter activity summarized across conditions. Condition must be provided.
`by`	A character vector. The assay to visualize the heatmap for. One of promoterCounts, normalizedPromoterCounts, absolutePromoterActivity and geneExpression (unambiguous substrings can be supplied). Defaults to absolutePromoterActivity.
`features`	Features to visualize. Either a list of promoterIds or geneIds. The features must correspond to the assay is used, i.e., if promoter assays are used, features must be promoterIds, while if gene expression assay is used, features must be geneIds. Defaults to NULL (visualizes all features of the assay).
`cex.legend`	A numeric value. Legend size.
`cex.row`	A numeric value. Row label size.
`cex.col`	A numeric value. Column label size.
`row.margin`	A numeric value. Row margins.
`col.margin`	A numeric value. Column margins.
`col`	A vector of colours. Length should correspond to number of experimental conditions. Defaults to NULL.
`breaks`	A numeric vector. Breaks for heatmap plotting.
`palette`	A character vector. One of bluered, redblue, redgreen, greenred. Defaults to bluered.

Value

Displays heatmap.

Examples

```
files <- list.files(system.file('extdata/vignette/junctions',
                        package = 'proActiv'),
                        full.names = TRUE, pattern = 'replicate5')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
result <- proActiv(files = files,
                        promoterAnnotation  = promoterAnnotation,
                        condition = rep(c('A549', 'HepG2'), each=1),
                        fileLabels = NULL,
                        ncores = 1)
result <- result[complete.cases(assays(result)[[1]]),]
plotHeatmap(result)
```

`plotPCA` *Performs principal component analysis*

Description

Performs principal component analysis

Usage

```
plotPCA(
  result,
  by = "absolutePromoterActivity",
  main = NULL,
  col = NULL,
  alpha = 0.75,
  cex.size = 2
)
```

Arguments

`result`	A SummarizedExperiment object return by proActiv, with assays giving promoter counts and activity with gene expression stored as metadata. rowData contains promoter metadata and absolute promoter activity summarized across conditions. Condition must be provided.
`by`	A character vector. The assay to perform principal component analysis by. One of promoterCounts, normalizedPromoterCounts, absolutePromoterActivity and geneExpression (unambiguous substrings can be supplied). Defaults to absolutePromoterActivity.
`main`	A character vector. Plot title (optional). Defaults to NULL.
`col`	A vector of colours. If NULL, uses standard ggplot colours. Defaults to NULL.
`alpha`	A numeric value in between 0 and 1. Determines point transparency.
`cex.size`	A numeric value. Determines point size.

Value

PCA plot.

Examples

```
files <- list.files(system.file('extdata/vignette/junctions',
                        package = 'proActiv'),
                        full.names = TRUE, pattern = 'replicate5')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
result <- proActiv(files = files,
                        promoterAnnotation  = promoterAnnotation,
                        condition = rep(c('A549', 'HepG2'), each=1),
                        fileLabels = NULL,
                        ncores = 1)
result <- result[complete.cases(assays(result)[[1]]),]
plotPCA(result)
```

`preparePromoterAnnotation`
Prepares promoter annotation from a gtf or txdb

Description

Prepares promoter annotation from a gtf or txdb

Usage

```
preparePromoterAnnotation(txdb, file, species)
```

Arguments

`txdb`	A txdb object. The txdb of the annotation version for which promoters will be identified. Either ‘txdb‘ or ‘file‘ argument must be specified, but not both.
`file`	A character object. The file path to a gtf/gff or txdb of the annotation version for which promoters will be identified. Either ‘txdb‘ or ‘file‘ argument must be specified, but not both.
`species`	A character object. The genus and species of the organism to be used in keepStandardChromosomes(). Supported species can be seen with names(genomeStyles()).

Value

A PromoterAnnotation object. The annotated intron ranges, promoter coordinates and the promoter id mapping are attributes of the promoter annotation data.

Examples

```
txdbPath <- system.file('extdata/vignette/annotations/',
                             'gencode.v34.annotation.subset.sqlite',
                              package = 'proActiv')
txdb <- AnnotationDbi::loadDb(txdbPath)
promoterAnnotation <- preparePromoterAnnotation(txdb = txdb,
                                                   species = 'Homo_sapiens')
```

`proActiv` *Estimates promoter counts and activity in a single command*

Description

Estimates promoter counts and activity in a single command

Usage

```
proActiv(
  files,
  promoterAnnotation,
  fileLabels = NULL,
  condition = NULL,
  genome = NULL,
  ncores = 1
)
```

Arguments

`files`	A character vector. The list of input files for which the junction read counts will be calculated
`promoterAnnotation`	A PromoterAnnotation object containing the intron ranges, promoter coordinates and the promoter id mapping
`fileLabels`	A character vector. The labels of input files for which the junction read counts will be calculated. These labels will be used as column names for each output data.frame object. If not provided, filenames will be used as labels. Defaults to NULL
`condition`	A character vector. The condition to which each sample belong to. Must correspond to the order of the files. If supplied, results are summarized by condition. Defaults to NULL
`genome`	A character. Genome version. Must be specified if input file type is a BAM file. Defaults to NULL
`ncores`	A numeric value. The number of cores to be used for counting junction reads. Defaults to 1 (no parallelization). This parameter will be used as an argument to BiocParallel::bplapply

Value

A SummarizedExperiment object with assays giving promoter counts and activity with gene expression. rowData contains promoter metadata and absolute promoter activity summarized across conditions (if condition is provided)

Examples

```
files <- list.files(system.file('extdata/vignette/junctions',
                      package = 'proActiv'),
                      full.names = TRUE, pattern = 'replicate5')
promoterAnnotation <- promoterAnnotation.gencode.v34.subset
result <- proActiv(files = files,
                      promoterAnnotation  = promoterAnnotation,
                      condition = rep(c('A549', 'HepG2'), each=1),
                      fileLabels = NULL,
                      ncores = 1)
```

PromoterAnnotation-class
S4 class for promoter annotation data for a specific annotation version

---

Description

S4 class for promoter annotation data for a specific annotation version

Usage

```
PromoterAnnotation(
  intronRanges = GenomicRanges::GRanges(),
  promoterIdMapping = data.frame(),
  promoterCoordinates = GenomicRanges::GRanges()
)

intronRanges(x)

## S4 method for signature 'PromoterAnnotation'
intronRanges(x)

promoterIdMapping(x)

## S4 method for signature 'PromoterAnnotation'
promoterIdMapping(x)

promoterCoordinates(x)

## S4 method for signature 'PromoterAnnotation'
promoterCoordinates(x)

intronRanges(x) <- value

## S4 replacement method for signature 'PromoterAnnotation'
intronRanges(x) <- value

promoterIdMapping(x) <- value

## S4 replacement method for signature 'PromoterAnnotation'
promoterIdMapping(x) <- value

promoterCoordinates(x) <- value

## S4 replacement method for signature 'PromoterAnnotation'
promoterCoordinates(x) <- value
```

Arguments

`intronRanges`	A GRanges object containing annotated intron ranges

`promoterIdMapping`	A data.frame containing mapping between transcript, TSS, promoter and gene ids
`promoterCoordinates`	A GRanges object containing promoter coordinates
`x`	A PromoterAnnotation object
`value`	intronRanges, promoterIdMapping or promoterCoordinates to be assigned

Value

A promoter annotation object with three slots: intronRanges, promoterIdMapping and promoter Coordinates

Functions

- `intronRanges`: Getter for intronRanges
- `intronRanges,PromoterAnnotation-method`: Getter for intronRanges
- `promoterIdMapping`: Getter for promoterIdMapping
- `promoterIdMapping,PromoterAnnotation-method`: Getter for promoterIdMapping
- `promoterCoordinates`: Getter for promoterCoordinates
- `promoterCoordinates,PromoterAnnotation-method`: Getter for promoterCoordinates
- `intronRanges<-`: Setter for intronRanges
- `intronRanges<-,PromoterAnnotation-method`: Setter for intronRanges
- `promoterIdMapping<-`: Setter for promoterIdMapping
- `promoterIdMapping<-,PromoterAnnotation-method`: Setter for promoterIdMapping
- `promoterCoordinates<-`: Setter for promoterCoordinates
- `promoterCoordinates<-,PromoterAnnotation-method`: Setter for promoterCoordinates

Slots

`intronRanges` A GRanges object. The intron ranges annotated with the promoter information.

`promoterIdMapping` A data.frame object. The id mapping between transcript ids, names, TSS ids, promoter ids and gene ids.

`promoterCoordinates` A GRanges object. Promoter coordinates (TSS) with gene id and internal promoter state

Examples

```
promoterAnnotation <- PromoterAnnotation()
intronRanges(promoterAnnotation) <- intronRanges(
                                    promoterAnnotation.gencode.v34.subset)
promoterIdMapping(promoterAnnotation) <- promoterIdMapping(
                                    promoterAnnotation.gencode.v34.subset)
promoterCoordinates(promoterAnnotation) <- promoterCoordinates(
                                    promoterAnnotation.gencode.v34.subset)
```

`promoterAnnotation.gencode.v34.subset`
Promoter annotation for Gencode.v34 (subset)

Description

Promoter annotation for Gencode.v34 (chr1:10,000,000 - 30,000,000)

Usage

```
promoterAnnotation.gencode.v34.subset
```

Format

A PromoterAnnotation (S4 Class) object containing all promoter annotation objects for Gencode.v34 chr1:10,000,000-30,000,000. The object has 3 slots:

intronRanges A GRanges object of 4,523 ranges corresponding to introns, annotated with the associated transcript.

promoterIdMapping The id mapping between transcript names, promoter ids and gene ids for Gencode v34.

promoterCoordinates A GRanges object of 1,380 ranges showing the tss coordinate for each promoter of Gencode v34 chr1:10,000,000-30,000,000, annotated with the associated gene id, coordinate of the 3' end of the first reduced exon, and intron id.

`reexports` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

AnnotationDbi `loadDb`

dplyr `%>%`, `arrange`, `as_tibble`, `filter`, `group_by`, `mutate`, `n`

SummarizedExperiment `assays`, `colData`, `rowData`

tibble `rownames_to_column`

Index