Package ‘consensusDE’

October 16, 2019

Type Package

Title RNA-seq analysis using multiple algorithms

Version 1.2.1

Author Ashley J. Waardenberg

Maintainer Ashley J. Waardenberg <a.waardenberg@gmail.com>

Description This package allows users to perform DE analysis using multiple algorithms. It seeks consensus from multiple methods. Currently it supports ``Voom'', ``EdgeR'' and ``DESeq''. It uses RUV-seq (optional) to remove unwanted sources of variation.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

biocViews Transcriptomics, MultipleComparison, Clustering, Sequencing, Software

Depends R (>= 3.5), BiocGenerics

Imports airway, AnnotationDbi, BiocParallel, Biobase, Biostrings, data.table, dendextend, DESeq2 (>= 1.20.0), EDASeq, ensembldb, edgeR, EnsDb.Hsapiens.v86, GenomicAlignments, GenomicFeatures, limma, org.Hs.eg.db, pcaMethods, RColorBrewer, Rsamtools, RUVSeq, S4Vectors, stats, SummarizedExperiment, TxDb.Dmelanogaster.UCSC.dm3.ensGene, utils

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

git_branch RELEASE_3_9

git_last_commit ad714ea

git_last_commit_date 2019-07-11

Date/Publication 2019-10-15

R topics documented:

`buildSummarized` *Generate summarized Read File for DE analyses*

Description

This function will create a summarized experiment, decribing reads from RNA-seq experiments that overlap a set of transcript features. Transcript features can be described as a gtf formatted table that is imported, or using a txdb. The summarized experiment can be build directly from bam files or by reading in counts in htseq format. This is designed to be straightforward and with minimised parameters for batch style RNA-seq analyses.

Usage

```
buildSummarized(sample_table = NULL, bam_dir = NULL,
  htseq_dir = NULL, gtf = NULL, tx_db = NULL,
  technical_reps = FALSE, map_reads = "transcript",
  mapping_mode = "Union", read_format = NULL, ignore_strand = FALSE,
  fragments = TRUE, summarized = NULL, output_log = NULL,
  filter = FALSE, BamFileList_yiedsize = NA_integer_, n_cores = 1,
  force_build = FALSE, verbose = FALSE)
```

Arguments

`sample_table`	A data.frame describing samples. For paired mode it must at least 2 columns, "file", "group", and option additional columns, "pairs" and "tech_replicate" for describing sample pairing and instances of technical replicates. The filename "file" must correspong to the name of the file in the directory supplied with the "bam_dir" parameter below - or ar error will be reported and buildSummarized will halt. This is not required if an existing summarized file is provided. Default = NULL
`bam_dir`	Full path to location of bam files listed in the "file" column in the sample_table provided above. This is not required if an existing summarized file is provided. Default = NULL
`htseq_dir`	Full path to location of htseq files listed in the "file" column in the sample_table described above. This is not required if an existing summarized file is provided. Files must end in ".txt". Default = NULL
`gtf`	Full path to a gtf file describing the transcript coordinates to map the RNA-seq reads to. GTF file is not required if providing a pre-computed summarized experiment file previously generated using buildSummarized() OR a tx_db object (below). Default = NULL
`tx_db`	An R txdb object. E.g. TxDb.Dmelanogaster.UCSC.dm3.ensGene. Default = NULL
`technical_reps`	Are there technical replicates to merge counts? I.e. are there multiple technical replicates run accross multiple lanes/sequencing runs. If "TRUE", unique sample names should be provided in a "tech_replicate" column of the "sample_table" for identification. Options are "TRUE" or "FALSE". Default = "FALSE"

`map_reads`	Which features to count reads by. Options are "transcript", "exon" or "cds". This will invoke transcriptsBy(), exonsBy() or cdsBy() respectively. Default = "transcript"
`mapping_mode`	Options are "Union", "IntersectionStrict" and "IntersectionNotEmpty". see "mode" in ?summarizeOverlaps for explanation. Default = "Union"
`read_format`	Are the reads from single-end or paired-end data? Option are "paired" or "single". An option must be selected if htseq_dir is NULL and read are summarized from BAM files. Default = NULL
`ignore_strand`	Ignore strand when mapping reads? see "ignore_strand" in ?summarizeOverlaps for explanation. Default = FALSE
`fragments`	When mapping_mode="paired", include reads from pairs that do not map with their corresponding pair? see "fragments" in ?summarizeOverlaps for explanation. Default = TRUE
`summarized`	Full path to a summarized experiment file. If buildSummarized() has already been performed, the output summarized file, saved in "/output_log/se.R" can be used as the input (e.g. if filtering is to be done). Default = NULL
`output_log`	Full path to directory for output of log files and saved summarized experiment generated.
`filter`	Perform filtering of low count and missing data from the summarized experiment file? This uses default filtering via "filterByExpr". See ?filterByExpr for further information. Default = FALSE
`BamFileList_yiedsize`	If running into memory problems. Set the number of lines to an integer value. See "yieldSize" description in ?BamFileList for an explanation.
`n_cores`	Number of cores to utilise for reading in Bam files. Use with caution as can create memory issues if BamFileList_yiedsize is not parameterised. Default = 1
`force_build`	If the sample_table contains less than two replicates per group, force a summarizedExperiment object to be built. Otherwise buildSummarized will halt. Default = FALSE.
`verbose`	Verbosity ON/OFF. Default=FALSE

Value

A summarized experiment

Examples

```
## Extract summarized following example in the vignette
## The example below will return a summarized experiment
## tx_db is obtained from TxDb.Dmelanogaster.UCSC.dm3.ensGene library
library(TxDb.Dmelanogaster.UCSC.dm3.ensGene)
## bam files are obtained from the GenomicAlignments package

## 1. Build a sample table that lists files and groupings
## - obtain list of files
file_list <- list.files(system.file("extdata", package="GenomicAlignments"),
                        recursive = TRUE,
                        pattern = "*bam$",
                        full = TRUE)
bam_dir <- as.character(gsub(basename(file_list)[1], "", file_list[1]))
```

```
## - create a sample table to be used with buildSummarized() below
## must be comprised of a minimum of two columns, named "file" and "group",
sample_table <- data.frame("file" = basename(file_list),
                           "group" = c("treat", "untreat"))

summarized_dm3 <- buildSummarized(sample_table = sample_table,
                                  bam_dir = bam_dir,
                                  tx_db = TxDb.Dmelanogaster.UCSC.dm3.ensGene,
                                  read_format = "paired",
                                  force_build = TRUE)
```

`diag_plots` *QC/diagnostic plotting*

Description

Wrappers for a series of plots to be used as diagnostics in RNA-seq analyses. Currently 10 plots are possible using this function: 1) Mapped reads, 2) Relative Log Expression (RLE), 3) Principle Component Analyis (PCA), 4) Residuals from a batch correction model, e.g. RUVseq, 5) Hierarchical clustering, 6) Densitiy distributions, 7) Boxplots, 8) MA plots, 9) Volcano Plots and 10) P-value distribution plots. Plots 1 to 6 utilise a "SeqExpressionSet" object for extracting information to plot. Plots 8-10 utilised a simple list class, containing all the data.frames of each comparison performed. See descriptions of each in the parameter options below and for format specification. See vignette for more information and examples.

Usage

```
diag_plots(se_in = NULL, merged_in = NULL, write = FALSE,
  plot_dir = NULL, legend = TRUE, label = TRUE, name = NULL,
  mapped_reads = FALSE, rle = FALSE, pca = FALSE,
  residuals = FALSE, hclust = FALSE, density = FALSE,
  boxplot = FALSE, ma = FALSE, volcano = FALSE, p_dist = FALSE)
```

Arguments

`se_in`	A "SeqExpressionSet" object or "RangedSummarizedExperiment" generated using "buildSummarized()". If the input is a "SeqExpressionSet", ensure that it included groups to be analysed. E.g. accessible as "se_in$group. Groupings are used to automate colouring of samples in unsupervised analyses. Default = NULL
`merged_in`	A data.frame that contains the merged results which are included in the outputs from multi_de_pairs(). These contain the ouputs from the pair-wise comparisons which allows plotting of MA, Volcano and p-value distributions. Where the outputs of multi_de_pairs() are to be used as inputs into diag_plots(), use multi_de_pairs()$merged as inputs. See example below. Default = NULL
`write`	Write the results to a pdf file? Options: TRUE, FALSE. This is to be used together with "plot_dir" and "write" parameters (below). Will report an error and halt if is TRUE and "plot_dir" and "write" are NULL. Default = FALSE
`plot_dir`	If "write" is TRUE, where to write the files to? The directory must already exist. E.g. "/path/to/my/pretty/plots/". Default = NULL

`legend`	Include legend in plots? Legend is based on group data in se_in. Options: TRUE, FALSE. Default = FALSE
`label`	Include point labels in plots? Points are based on ID column from merged_in. Options: TRUE, FALSE. Default = FALSE
`name`	If "write" is TRUE, what to name the plot? The file name will always be preceded with "QC_" and end in ".pdf". E.g. name="very_pretty_plots" will produce a file named "QC_very_pretty_plots.pdf" in "/path/to/my/pretty/plots/". Default = NULL
`mapped_reads`	Plot mapped reads per sample as a barchart. Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`rle`	Plot Relative Log Expressio (RLE) of samples for assessment of sample quality. See ?plotRLE for further details. Requires se_in to be a "SeqExpressionSet"and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`pca`	Perform unsupervised Principle Component Analysis (PCA) and plot results. By default performs Singular Value Decomposition. Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`residuals`	If RUV-seq has been applied to dataset, plot the residuals identified in the model. Only works for one set of residuals. Data is also accessible using pData(se_in)$W_1. Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`hclust`	Performs unsupervised hierarchical clustering of samples. Colours sample below plot according to group and numbered by inputs. Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`density`	Plot density distributions of log2(count-per-million). Will automatically extract normalised counts over non-normalised counts is available in "SeqExpressionSet". Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`boxplot`	Boxplot of density distributions of log2(count-per-million). Will automatically extract normalised counts over non-normalised counts is available in "SeqExpressionSet". Requires se_in to be a "SeqExpressionSet" and utilise "group" meta-data for colouring. Options: TRUE, FALSE. Default = FALSE
`ma`	Plot Mean versus. Log2 Fold-Change of comparison. Requires a data.frame as input to "merged_in" with the following column names "ID", "AvExpr", "Log2FC" and "Adj_PVal".The data frame should be sorted, as the top 10 in the table are also plotted. Options: TRUE, FALSE. Default = FALSE
`volcano`	Volcano plot of Log2 Fold-Change and significance of comparison. Requires a data.frame as input to "merged_in" with the following column names "ID", "AvExpr", "Log2FC" and "Adj_PVal". The data frame should be sorted, as the top 10 in the table are also plotted. Options: TRUE, FALSE. Default = FALSE
`p_dist`	P-value distribution plot. Requires a data.frame as input to "merged_in" with the following column names "ID", "AvExpr", "Log2FC" and "Adj_PVal". The data frame should be sorted, as the top 10 in the table are also plotted. Options: TRUE, FALSE. Default = FALSE

Value

Returns pretty plots

Examples

```
## Load the example data set and attach
## The example below will display a PCA plot before normalisation
library(airway)
data(airway)
## Name the groups of the data.
colData(airway)$group <- colData(airway)$dex
## Identify the file locations
colData(airway)$file <- rownames(colData(airway))
## Filter low count data:
airway_filter <- buildSummarized(summarized = airway,
                                 filter = TRUE)
## for illustration, use random sample of 1000 transcripts
set.seed(1234)
airway_filter <- sample(airway_filter, 1000)
## The following is example code to perform a PCA plot
## see vignette for more details of displaying each plot
## diag_plots(se_in = airway_filter,
##           name = "airway example data",
##           pca = TRUE)
```

`multi_de_pairs` *Batch - multiDE analysis of many comparisons*

Description

Given a summarized experiment generated using buildSummarized() this function will automatically perform differential expression (DE) analysis for all possible groups using 3 different methods 1) EdgeR, 2) Voom and 3) DEseq2. It will also output 10x diagnostic plots automatically, if the plotting options are selected (see ?diag_plots for more details).

Usage

```
multi_de_pairs(summarized = NULL, paired = "unpaired",
  intercept = NULL, adjust_method = "BH", EDASeq_method = "upper",
  norm_method = "EDASeq", ruv_correct = FALSE,
  ensembl_annotate = NULL, gtf_annotate = NULL, plot_dir = NULL,
  output_voom = NULL, output_edger = NULL, output_deseq = NULL,
  output_combined = NULL, verbose = FALSE, legend = TRUE,
  label = TRUE)
```

Arguments

`summarized`	A "RangedSummarizedExperiment" object with included groups to be analysed. For format specifications see ?buildSummarized. E.g. accessible as "summarized$group". Groups are used to automate colouring of samples in unsupervised analyses. Default = NULL

`paired`	Are the sample paired? If "paired" a paired statistical analysis by including factors as pairs described in the "pairs" column of the "RangedSummarizedExperiment" object in the model (accessible as summarized$pairs). Options are "unpaired" or "paired". Default="unpaired"
`intercept`	Optional ability to set the base term for fitting the model. This is not necessary as all pairs are computed automatically. The base term, if set, must match the name of s group in "summarized$group". Default = NULL
`adjust_method`	Method used for multiple comparison adjustment of p-values. Options are: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" or "none". See ?p.adjust.methods for a details description and references. Default = "BH"
`EDASeq_method`	Method for normalisation (applies to QC results using EDASeq and RUV when EDASeq is selected). Options are:"median","upper","full". Default = "upper"
`norm_method`	Methods for normalisation. Options are: "EDASeq" or "all_defaults". When "all_defaults" is selected, this will use all default normalisation methods for differential expression, EDASeq for QC, and edgeR "upperquantile" for determining RUV residuals (as per RUVSeq vignette). When "EDASeq" is selected, this will use EDASeq normalisation throughout. EDASeq normalisation method is selected using "EDASeq_method". Default = "EDASeq".
`ruv_correct`	Remove Unwanted Variation (RUV)? See ?RUVr for description. Currently only RUVr, which used the residuals is enabled and one factor of variation is determined. If set to TRUE and a "plot_dir" is provided, additional plots after RUV correction and the RUV residuals will be reported. Residuals are obtained through fitting a generalised linear model (GLM) using EdgeR. Residuals are then incorporated into the SummarizedExperiment object and all models for DE analysis. Options = TRUE, FALSE. Default = FALSE.
`ensembl_annotate`	If the dataset has been mapped to ensembl transcript identifiers, obtain additional annotation of the ensembl transcripts. A R Genome Wide Annotation object e.g. org.Mm.eg.db for mouse or org.Hs.eg.db for human must be provided. Default = NULL
`gtf_annotate`	Full path to a gtf file describing the transcripts. If provided will obtain gene symbols from gtf file. If a ensembl_annotate object is also provided, this will extract annotations based on the symbols extracted from the GTF file. It is recommended to provide both a gtf file and a tx_db for better annotation results. Default = NULL
`plot_dir`	Full path to directory for output of plots (pdf files). See ?diag_plots for more details. Default = NULL
`output_voom`	If you wish to output the results of the Voom analysis, provide a full path to directory for output of files. Default = NULL
`output_edger`	If you wish to output the results of the EdgeR analysis, provide a full path to directory for output of files. Default = NULL
`output_deseq`	If you wish to output the results of the DEseq2 analysis, provide a full path to directory for output of files. Default = NULL
`output_combined`	consensusDE will report the results of Voom, EdgeR and DEseq2 as a combined report. If you wish to output the results of the COMBINED analysis, provide a full path to directory for output of files. In addition to the combined data, it will also output the raw count and normalised data to the same directory. Default = NULL

`verbose`	Verbosity ON/OFF. Default=FALSE
`legend`	Include legend in plots? Legend is based on group data in summarized Options: TRUE, FALSE. Default = TRUE
`label`	Include point labels in plots? Points are based on ID column after DE analysis from merged results. Options: TRUE, FALSE. Default = TRUE

Value

A list of all the comparisons conducted. ## See vignette for more details.

Examples

```
## Load the example data set and attach - see vignette for more details
## The example below will perfrom DE analysis on all pairs of data
library(airway)
data(airway)
## Name groups of the data.
colData(airway)$group <- colData(airway)$dex
## Identify file locations
colData(airway)$file <- rownames(colData(airway))
#' ## Filter low count data:
airway_filter <- buildSummarized(summarized = airway,
                                filter = TRUE)
## for illustration, we only use random sample of 1000 transcripts
set.seed(1234)
airway_filter <- sample(airway_filter, 1000)
## Run multi_de_pairs() with-out RUV correction
## To run with RUV correction, use ruv_correct = TRUE
all_pairs_airway <- multi_de_pairs(summarized = airway_filter,
                                  ruv_correct = FALSE,
                                  paired = "unpaired")
```

Index