Package ‘GREP2’

May 8, 2018

Title GEO RNA-seq Experiments Processing Pipeline

Version 0.0.0.99

Author Naim Mahi [cre,aut], Mario Medvedovic [aut, ctb]

Maintainer Naim Mahi <naim.al.mahi@gmail.com>

Description An R based comprehensive pipeline to download and process GEO RNA-seq data.

Depends R (>= 3.4.1)

Imports XML, rentrez, RCurl, GEOquery, Biobase, parallel, tximport, EnsDb.Hsapiens.v86, EnsDb.Rnorvegicus.v79, EnsDb.Mmusculus.v79, AnnotationDbi, org.Hs.eg.db, org.Mm.eg.db, org.Rn.eg.db, utils, GenomicFeatures

Suggests knitr, BiocStyle, RMySQL, ensembldb, rmarkdown

License GPL-3

VignetteBuilder rmarkdown

LazyData true

biocViews GEO, RNASeq, GeneExpression, Software, DataImport, Preprocessing, QualityControl, Alignment

RoxygenNote 6.0.1

R topics documented:

build_index *Build index for mapping using Salmon*

Description

`build_index` for mapping reads using Salmon.

Usage

```
build_index(species = c("human", "mouse", "rat"), kmer = 31, destdir,
  ens_release = 92)
```

Arguments

`species` name of the species. Only `'human'`, `'mouse'`, and `'rat'` are allowed to use.

`kmer` k-mer size for indexing. default is 31. See `'Salmon'` for details.

`destdir` directory to save index files.

`ens_release` version of Ensembl release.

References

Rob Patro, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford (2017): Salmon provides fast and bias-aware quantification of transcript expression. Nature methods, 14(4), 417. `https://www.nature.com/articles/nmeth.4197`

Examples

```
build_index(species="human", kmer=31, destdir=".", ens_release=92)
```

get_fastq *Download fastq files*

Description

`get_fastq` downloads fastq files using SRA toolkit. We recommend using Aspera for fast downloading. You need to install Aspera(`http://www.asperasoft.com/`) for using `ascp` option.

Usage

```
get_fastq(srr_id, library_layout = c("SINGLE", "PAIRED"),
  get_sra_file = FALSE, sra_files_dir = NULL, n_thread, destdir)
```

Arguments

`srr_id` SRA run accession ID.

`library_layout` layout of the library used. Either `'SINGLE'` or `'PAIRED'`.

`get_sra_file` logical, whether to download SRA file first and get fastq files afterwards or directly download fastq files.

`sra_files_dir` directory where SRA files are saved. If you use `get_sra_file=FALSE` then `sra_files_dir=NULL`.

`n_thread` number of cores to use.

`destdir` directory where all the results will be saved.

Value

A single fastq file will be generated for SINGLE end reads and two files for PAIRED end reads.

Examples

```
get_fastq(srr_id="SRR6324192", library_layout="SINGLE", get_sra_file=FALSE,
sra_files_dir=NULL, n_thread=2, destdir=".")
```

`get_metadata` *Download metadata from GEO and SRA*

Description

Download metadata from GEO and SRA

Usage

```
get_metadata(geo_series_acc)
```

Arguments

`geo_series_acc` GEO series accession ID.

Value

a list of GEO and SRA metadata.

Examples

```
get_metadata(geo_series_acc="GSE107363")
```

get_srr *Download SRA run files*

Description

`get_srr` downloads SRA files using Aspera (`http://www.asperasoft.com/`) or FTP. We recommend using Aspera for fast downloading. You need to install Aspera for using `ascp` option.

Usage

```
get_srr(srr_id, destdir, ascp = TRUE, prefetch_workspace, ascp_path)
```

Arguments

`srr_id` SRA run accession ID.

`destdir` directory where all the results will be saved.

`ascp` logical, whether to use Aspera for downloading SRA files.

`prefetch_workspace` directory where SRA run files will be downloaded. This parameter is needed if `ascp=TRUE`. The location of this directory can be found by going to the aspera directory (/.aspera/connect/bin/) and typing `'vdb-config -i'`. A new window will pop-up and under the `'Workspace Name'`, you will find the location. Usually the default is `'/home/username/ncbi/public'`.

`ascp_path` path to the Aspera software.

Value

SRA run accession file with extension ".sra". If you use `ascp=TRUE`, then downloaded files will be saved under `'/prefetch_workspace/sra'` directory. If `ascp=FALSE`, then files will be saved in the `'destdir'`

Examples

```
get_srr(srr_id="SRR6324192", destdir=".", ascp=TRUE,
prefetch_workspace="path_to_prefetch_workspace", ascp_path="path_to_aspera")
```

`process_geo_rnaseq` *A complete pipeline to process GEO RNA-seq data*

Description

`process_geo_rnaseq` downloads and processes GEO RNA-seq data for a given GEO series accession ID. It filters metadata for RNA-seq samples only. We use SRA toolkit for downloading SRA data, Trimmomatic for read trimming (optional), and Salmon for read mapping.

Usage

```
process_geo_rnaseq(geo_series_acc, destdir, ascp = TRUE, prefetch_workspace,
  ascp_path, get_sra_file = FALSE, trim_fastq = FALSE,
  trimmomatic_path = NULL, index_dir, species = c("human", "mouse", "rat"),
  countsFromAbundance = c("no", "scaledTPM", "lengthScaledTPM"),
  n_thread = 2)
```

Arguments

`geo_series_acc`
GEO series accession ID.

`destdir` directory where all the results will be saved.

`ascp` logical, whether to use Aspera connect to download SRA run files. If FALSE, then wget will be used to download files which might be slower than `'ascp'` download.

`prefetch_workspace`
directory where SRA run files will be downloaded. This parameter is needed when `ascp=TRUE`. The location of this directory can be found by going to the aspera directory (/.aspera/connect/bin/) and typing `'vdb-config -i'`. A new window will pop-up and under the `'Workspace Name'`, you will find the location. Usually the default is `'/home/username/ncbi/public'`.

`ascp_path` path to the Aspera software.

`get_sra_file` logical, whether to download SRA file first and get fastq files afterwards.

`trim_fastq` logical, whether to trim fastq file.

`trimmomatic_path`
path to Trimmomatic software.

`index_dir` directory of the indexing files needed for read mapping using Salmon. See build_index.

`species` name of the species. Only `'human'`, `'mouse'`, and `'rat'` are allowed to use.

`countsFromAbundance`
whether to generate counts based on abundance. Available options are: `'no'`, `'scaledTPM'` (abundance based estimated counts scaled up to library size), `'lengthScaledTPM'` (default, scaled using the average transcript length over samples and library size). See Bioconductor package tximport for further details.

n_thread number of cores to use.

Value

a list of metadata from GEO and SRA saved in the destdir. Another list of gene and transcript level estimated counts summarized by Bioconductor package 'tximport' is also saved in the destdir.

References

Rob Patro, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford (2017): Salmon provides fast and bias-aware quantification of transcript expression. Nature methods, 14(4), 417. https://www.nature.com/articles/nmeth.4197

Charlotte Soneson, Michael I. Love, Mark D. Robinson (2015): Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research. http://dx.doi.org/10.12688/f1000research.7563.1

Philip Ewels, Mans Magnusson, Sverker Lundin, and Max Kaller (2016): MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics, 32(19), 3047-3048. https://doi.org/10.1093/bioinformatics/btw354

Examples

```
process_geo_rnaseq (geo_series_acc="GSE107363", destdir=".", ascp=TRUE,
prefetch_workspace="path_to_prefetch_workspace",ascp_path="path_to_aspera",
get_sra_file=FALSE, trim_fastq=FALSE, trimmomatic_path=NULL,index_dir="path_to_indexDir",
species="human", countsFromAbundance = "lengthScaledTPM", n_thread=2)
```

---

run_fastqc *QC report for each fastq files using FastQC*

Description

run_fastqc HTML report of each fastq files using FastQC. You need to install FastQC from https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Usage

```
run_fastqc(destdir, fastq_dir, n_thread)
```

Arguments

destdir	directory where all the results will be saved.
fastq_dir	directory of the fastq files.
n_thread	number of cores to use.

Value

HTML report of the fastq files under fastqc directory.

Examples

```
run_fastqc(destdir=".", fastq_dir="path_to_fastq_dir", n_thread=2)
```

`run_multiqc` *Generate combined QC report for Salmon and FastQC*

Description

`run_fastqc` generates a single HTML report from the fastQC reports and salmon read mapping results using MultiQC.

Usage

```
run_multiqc(fastqc_dir, salmon_dir, destdir)
```

Arguments

`fastqc_dir` directory where all the FastQC files are saved.

`salmon_dir` directory of the salmon files.

`destdir` directory where you want to save the combined QC report.

Value

HTML report.

References

Philip Ewels, Mans Magnusson, Sverker Lundin, and Max Kaller (2016): MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics, 32(19), 3047-3048. `https://doi.org/10.1093/bioinformatics/btw354`

Examples

```
run_fastqc(destdir=".", fastq_dir="path_to_fatsq_dir", n_thread=2)
```

run_salmon *Quantify transcript abundances using Salmon*

Description

run_salmon is a wrapper function for mapping reads to quantify transcript abundances using Salmon. You need to install Salmon and build index to run this function. For index building see build_index.

Usage

```
run_salmon(srr_id, library_layout = c("SINGLE", "PAIRED"), index_dir, destdir,
  fastq_dir, use_trimmed_fastq = FALSE, other_opts = NULL, n_thread)
```

Arguments

srr_id SRA run accession ID.

library_layout layout of the library used. Either 'SINGLE' or 'PAIRED'.

index_dir directory of the indexing files needed for read mapping using Salmon. See build_index.

destdir directory where all the results will be saved.

fastq_dir directory of the fastq files.

use_trimmed_fastq logical, whether to use trimmed fastq files.

other_opts Other options to use. See Salmon documentation for the available options.

n_thread number of cores to use.

Details

run_salmon We use default options of Salmon. This function works for a single sample. You can use this function in a loop for multiple samples. For other options from Salmon use 'other_opts'.

Value

The following items will be returned and saved in the salmon directory:

1. quant_new.sf: plain-text, tab-separated quantification file that contains 5 column: Name,Length,EffectiveLength,TPM, and NumReads.
2. cmd_info.json: A JSON format file that records the main command line parameters with which Salmon was invoked for the run that produced the output in this directory.
3. aux_info: This directory will have a number of files (and subfolders) depending on how salmon was invoked.
4. meta_info.json: A JSON file that contains meta information about the run, including stats such as the number of observed and mapped fragments, details of the bias modeling etc.

5. ambig_info.tsv: This file contains information about the number of uniquely-mapping reads as well as the total number of ambiguously-mapping reads for each transcript.
6. lib_format_counts.json: This JSON file reports the number of fragments that had at least one mapping compatible with the designated library format, as well as the number that didn't.
7. libParams: The auxiliary directory will contain a text file called flenDist.txt. This file contains an approximation of the observed fragment length distribution.

References

Rob Patro, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford (2017): Salmon provides fast and bias-aware quantification of transcript expression. Nature methods, 14(4), 417. `https://www.nature.com/articles/nmeth.4197`

Examples

```
run_salmon(srr_id="SRR6324192", library_layout="SINGLE", index_dir="path_to_index_dir",
destdir=".", fastq_dir="path_to_fastq_dir", use_trimmed_fastq=FALSE,
other_opts=NULL, n_thread=2)
```

`run_tximport` *Wrapper function to run tximport*

Description

`run_tximport` function runs tximport on transcript level abundances from Salmon to summarize to gene level. See Bioconductor package tximport for details.

Usage

```
run_tximport(srr_id, species = c("human", "mouse", "rat"), salmon_dir,
  countsFromAbundance = c("no", "scaledTPM", "lengthScaledTPM"))
```

Arguments

`srr_id` SRA run accession ID.

`species` name of the species. Only `'human'`, `'mouse'`, and `'rat'` are allowed to use.

`salmon_dir` directory where salmon files are saved.

`countsFromAbundance`
whether to generate counts based on abundance. Available options are: `'no'`, `'scaledTPM'` (abundance based estimated counts scaled up to library size), `'lengthScaledTPM'` (default, scaled using the average transcript length over samples and library size).

Details

We use Ensembl annotation for both genes and transcripts.

Value

a list of gene and transcript level estimated counts.

References

Charlotte Soneson, Michael I. Love, Mark D. Robinson (2015): Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research. `http://dx.doi.org/10.12688/f1000research.7563.1`

Examples

```
run_tximport(srr_id="SRR6324192", species="human", salmon_dir="path_to_salmon_files_dir",
countsFromAbundance = "lengthScaledTPM")
```

`trim_fastq` *Trim fastq files using Trimmomatic*

Description

`trim_fastq` trim fastq files based on the illumina instruments using Trimmomatic.

Usage

```
trim_fastq(srr_id, fastq_dir, instrument, trimmomatic_path,
  library_layout = c("SINGLE", "PAIRED"), n_thread)
```

Arguments

`srr_id`	SRA run accession ID.
`fastq_dir`	directory of the fastq files.
`instrument`	name of the illumina sequencing platform. For example, `'HiSeq'`.
`trimmomatic_path`	path to the Trimmomatic software.
`library_layout`	layout of the library used. Either `'SINGLE'` or `'PAIRED'`.
`n_thread`	number of cores.

Details

The following parameters are used as default in the trimmoatic function:

1. Remove leading low quality or N bases (below quality 3) (LEADING:3)
2. Remove trailing low quality or N bases (below quality 3) (TRAILING:3)
3. Scan the read with a 4-base wide sliding window, cutting when the average quality per base drops below 15 (SLIDINGWINDOW:4:15)
4. Drop reads below the 36 bases long (MINLEN:36)

Value

trimmed fastq files.

References

Anthony M. Bolger, Marc Lohse, and Bjoern Usadel (2014): Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics, 30(15), 2114-2120. `https://doi.org/10.1093/bioinformatics/btu170`

Examples

```
trim_fastq(srr_id="SRR6324192", fastq_dir=".", instrument="HiSeq",
trimmomatic_path="path_to_trimmomtic", library_layout="SINGLE", n_thread=2)
```