Package 'Rbowtie'

October 9, 2013

Type Package

Title R bowtie wrapper

Version 1.0.3

Date 2013-04-15

Author Florian Hahne, Anita Lerch, Michael B Stadler

Maintainer Michael Stadler <michael.stadler@fmi.ch>

Suggests parallel

Description This package provides an R wrapper around the popular bowtie short read aligner and around SpliceMap, a de novo splice junction discovery and alignment tool. The package is used by the QuasR bioconductor package. We recommend to use the QuasR package instead of using Rbowtie directly.

License Artistic-1.0 | file LICENSE

LazyLoad yes

biocViews HighThroughputSequencing

R topics documented:

Rbowtie-package *R bowtie wrapper*

Description

This package provides an R wrapper around the popular bowtie short read aligner and around SpliceMap, a de novo splice junction discovery and alignment tool.

The package is used by the QuasR bioconductor package. We recommend to use the QuasR package instead of using Rbowtie directly. The QuasR package provides a simpler interface then Rbowtie and covers the whole analysis workflow of typical ultra-high throughput sequencing experiments, starting from the raw sequence reads, over pre-processing and alignment, up to quantification.

Details

See packageDescription('Rbowtie') for package details.

Author(s)

Florian Hahne, Anita Lerch, Michael B Stadler

See Also

bowtie, SpliceMap

Examples

```
## Not run:
    example(bowtie)
    example(SpliceMap)

## End(Not run)
```

bowtie_build *Interface to bowtie*

Description

The following functions can be used to call the bowtie and bowtie-build binaries.

We recommend to use the QuasR package instead of using `bowtie` and `bowtie_build` directly. The QuasR package provides a simpler interface then Rbowtie and covers the whole analysis workflow of typical ultra-high throughput sequencing experiments, starting from the raw sequence reads, over pre-processing and alignment, up to quantification.

Usage

```
bowtie_build(references, outdir, ..., prefix = "index", force = FALSE,
strict = TRUE)

bowtie(sequences, index, ..., type = c("single", "paired", "crossbow"),
outfile, force = FALSE, strict = TRUE)

bowtie_build_usage()

bowtie_usage()

bowtie_version()
```

Arguments

references	Character vector. The path to the files containing the references for which to build a bowtie index.
outdir	Character scalar. The path to the output directory in which to store the bowtie index. If the directory already exists, the function function will cast an error, unless `force==TRUE`.
prefix	Character scalar. The prefix to use for the bowtie index files.
sequences	If `type` is either `single` or `crossbow`, a character vector of filenames if additional argument `c==FALSE`, otherwise a vector of read sequences. If `type` is `paired`, a list of filnames ore sequences of length 2, where the first list item corresponds to the fist mate pair sequences, and the second list item to the second mate pair sequences.
index	Character scalar. The path to the bowtie index and prefix to align against, in the form `</path/to/index>/<prefix>`.
type	Character scalar, one in `c("single", "paired", "crossbow")`. If `single`, the input sequences are interpreted as single reads. If `paired`, they are supposed to be mate pair reads and if `crossbow`, they are considered to be Crossbow-style reads.
outfile	Character scalar. A path to a files used for the alignment output. If missing, the alignments will be returned as a regular R character vector.
force	Logical. Force overwriting of `outdir` or `outfile`.
strict	Logical. Turn off strict checking of input arguments.
...	Additional arguments to be passed on to the binaries. See below for details.

Details

All additional arguments in `...` are interpreted as additional parameters to be passed on to the binaries. For flags, those are supposed to be logicals (e.g., `quiet=TRUE` will be translated into `--q`, `q=TRUE` in `-q`, and so on). Parameters with additional input are supposed to be character or numeric vectors, where the individual vector elements are collapsed into a single comma-separated string (e.g., `k=2` is translated into `k 2`, `bmax=100` into `--bmax 100`, `3=letters[1:3]` into `-3 a,b,c`, and so on). Note that some arguments to the bowtie binary will be ignored if they are already handled as

explicit function arguments. See the output of `bowtie_usage()` and `bowtie_build_usage()` for details about available parameters.

Value

The output generated by calling the binaries. For `bowtie_build` this is typically a report of the index generation, for `bowtie` this can be a vector of aligments (if `outfile` is missing), otherwise an empty character scalar.

`bowtie_usage()` and `bowtie_build_usage()` return the usage information for the respective binaries.

`bowtie_version()` return the bowtie versions information.

Author(s)

Florian Hahne

References

Langmead B, Trapnell C, Pop M, Salzberg SL. *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biology 10:R25.

Langmead B, Schatz M, Lin J, Pop M, Salzberg SL. *Searching for SNPs with cloud computing*. Genome Biology 10:R134.

Examples

```
td <- tempdir()

## Building a bowtie index
refs <- dir(system.file(package="Rbowtie", "samples", "refs"),
full=TRUE)
tmp <- bowtie_build(references=refs, outdir=file.path(td, "index"),
force=TRUE)
head(tmp)
dir(file.path(td, "index"))
tmp2 <- bowtie_build(references=refs, outdir=file.path(td,"indexColor"),
force=TRUE, C=TRUE)
dir(file.path(td, "indexColor"))
head(tmp2)


## Alignments
reads <- system.file(package="Rbowtie", "samples", "reads", "reads.fastq")
tmp <- bowtie(sequences=reads, index=file.path(td, "index", "index"))
tmp
bowtie(sequences=reads, index=file.path(td, "index", "index"),
outfile=file.path(td, "alignments.txt"), best=TRUE, force=TRUE)
readLines(file.path(td, "alignments.txt"))

bowtie(sequences=list("TGGGTGGGGTATTCTAGAAATTTCTATTAATCCT",
                      "TCTGTTCAAGTCAGATGGTCACCAATCTGAAGAC"),
index=file.path(td, "index", "index"), type="paired", c=TRUE)
```

SpliceMap *Interface to SpliceMap*

Description

The following function can be used to call the SpliceMap binaries.

We recommend to use the QuasR package instead of using SpliceMap directly. The QuasR package provides a simpler interface then Rbowtie and covers the whole analysis workflow of typical ultra-high throughput sequencing experiments, starting from the raw sequence reads, over pre-processing and alignment, up to quantification.

Usage

```
SpliceMap(cfg)
```

Arguments

cfg	A list containing named elements with SpliceMap parameters (see sQuoteDetails or SpliceMap documentation).

Details

The SpliceMap function performes the same steps as the runSpliceMap binary from the *SpliceMap* software package, but using R functions to improve compatibility on Windows.

While the original *SpliceMap* software package is able to call different tools to find sub-read alignments, the SpliceMap function from the Rbowtie package works only with bowtie contained in the package itself. Further modifications from the original version include the reporting of unmapped reads at the end of the output sam file, and the restriction to a single (pair) of input sequence file(s).

The cfg argument is a list with *SpliceMap* configuration parameters that would normally be specified using the *SpliceMap* configuration file. Here is a list of supported parameters extracted from the sample config file that is distributed with *SpliceMap*:

- genome_dir (single character value) Directory of the chromosome files in FASTA format, or path to a single FASTA file containing all chromosomes. If a directory, each chromosome can be in a separate file or chromosomes can be concatenated, ie. chr1.fa, chr2.fa, ...
- reads_list1 and reads_list2 (single character values) These are the two input sequence files. reads_list2 can be missing if reads are not paired-end. Note: reads_list1 must be the first pair, and pair-reads should be in the "forward-reverse" format.
- read_format (single character value) Format of the sequencer reads, also make sure reads are not split over multiple lines. Choices are: FASTA, FASTQ, RAW
- quality_format (single character value) Format of the quality string if FASTQ is used. Choices are:
  - phred-33 Phred base 33 (same as Sanger format)
  - phred-64 Phred base 64 (same as Illumina 1.3+)
  - solexa Format used by solexa machines

- `outfile` (single character value) Name of the output file (spliced alignments in SAM format). If a file with this name already exists, `SpliceMap` will stop with an exception. Note that unmapped reads will be appended to the end of the output file (for paired-end experiments, only pairs without alignments for any read are considered unmapped).
- `temp_path` (single character value) Directory name of the directory that stores temporary files. All temporary files will be created in a subfolder of `temp_path` and will be removed when `SpliceMap` finishes successfully or failed at an intermediate step.
- `max_intron` (single integer value) Maximum intron size, this is absolute 99th-percentile maximum. Introns beyond this size will be ignored. If you don't set this, we will assume a mamalian genome (400,000)
- `min_intron` (single integer value) 25-th intron size, this is the lower 25th-percentile intron size. This is not the smallest size that SpliceMap will search. That is about ~25bp. If you don't set this, we will assume a mammalian genome (20,000)
- `max_multi_hit` (single integer value) Maximum number of multi-hits. If a 25-mer seed has more than this many multi-hits, it will be discarded. Default is 10.
- `full_read_length` (single integer value) Full read length. SpliceMap will only use the first "full_read_length" bp for mapping. If the read is shorter than "full_read_length", the full read will be used before head clip. If you don't set this parameter, SpliceMap will use as many as possible. This is for the case where the reads might have N's at the end. It is always desireable to cut off the N's
- `head_clip_length` (single integer value) Number of bases to clip off the head of the read. This clipping is applied after "full_read_length"
- `seed_mismatch` (single integer value) Number of mismatches allowed in half-seeding. Choices are 0,1(default) or 2
- `read_mismatch` (single integer value) Maximum number of mismatches allowed in entire read. No limit on value, however SpliceMap can only identify reads with a maximum of 2 mismatches per 25bp. Default is 2.
- `max_clip_allowed` (single integer value) Maximum number of bases allowed to be soft clipped from the ends of reads during alignment. This is required as mismatches near junctions could cause parts of a a read to not map. Default is 40.
- `num_chromosome_together` (single integer value) Number of chromosomes to process at once, to take advantage of multi-core systems. The child processes are created using the `makeCluster` function from the **parallel** package. This is not threading, so it will take extra memory. However, running 2 at a time should be fine on current hardware. Default = 2
- `bowtie_base_dir` (single character value) Base of bowtie index, this should be the same genome as the chromosome files eg. if you bowtie files are "genome/hg18/genome.1.ewbt", ... then your base dir is "genome/hg18/genome"
- `num_threads` (single integer values) Number of threads to use for `bowtie` mapping. Default value is 2
- `try_hard` (single character value) Try hard? Choices are "yes" or "no". Default value is "yes" (about 15% slower).
- `selectSingleHit` (single logical value) If `TRUE` and multiple alignments are found for a read (pair), only a single alignment is selected randomly and reported. This is a new paramater only available in the `R` version of `SpliceMap`, but not in the original implementation that could report several alignments per read (pair).

The following parameters are mandatory:

- genome_dir
- reads_list1 (reads_list2 for paired-end experiments)
- read_format
- quality_format
- bowtie_base_dir
- outfile
- temp_path
- num_threads
- quality_format
- selectSingleHit

Value

An invisible character vector of length one with the file name of the generated output SAM file.

An exception is thrown if a failure is detected at one of the many steps.

Author(s)

Michael Stadler

References

Au KF, Jiang H, Lin L, Xing Y, Wong WH. *Detection of splice junctions from paired-end RNA-seq data by SpliceMap*. Nucleic Acids Research, 38(14):4570-8 (2010).

See Also

makeCluster from package **parallel**

Examples

```
## Building a bowtie index
refDir <- system.file(package="Rbowtie", "samples", "refs")
indexDir <- file.path(tempdir(), "refsIndex")

tmp <- bowtie_build(references=dir(refDir, full=TRUE), outdir=indexDir, prefix="index", force=TRUE)

## Alignments
readsFiles <- system.file(package="Rbowtie", "samples", "reads", "reads.fastq")
samFiles <- file.path(tempdir(), "splicedAlignments.sam")

cfg <- list(genome_dir=refDir,
            reads_list1=readsFiles,
            read_format="FASTQ",
            quality_format="phred-33",
            outfile=samFiles,
```

```
            temp_path=tempdir(),
            max_intron=400000,
            min_intron=20000,
            max_multi_hit=10,
            seed_mismatch=1,
            read_mismatch=2,
            num_chromosome_together=2,
            bowtie_base_dir=file.path(indexDir, "index"),
            num_threads=4,
            try_hard="yes",
            selectSingleHit=TRUE)

res <- SpliceMap(cfg)
```

Index