An Introduction to Rbowtie2

Zheng Wei and Wei Zhang

2017-08-24

MOE Key Laboratory of Bioinformatics and Bioinformatics Division,

TNLIST /Department of Automation, Tsinghua University

{wei-z14,w-zhang16}@mails.tsinghua.edu.cn

Introduction

The package provides an R wrapper of bowtie2 and AdapterRemoval. Bowtie2 is the popular sequencing reads aligner, which is good at aligning reads with length above 50bp[1]. AdapterRemoval is a convenient tool for rapid adapter trimming, identification, and read merging[2]. Both of them are implemented with C++. We use Rcpp package to wrap them into an R package that provide user friendly interfaces for R users.

You can preprocess the raw sequencing data by using AadapterRemoval even if adapter(s) information is missing. Then, bowtie2 can aligned these preprocessed reads to the references.

This package is developed and maintained by members of Xiaowo Wang Lab: http://bioinfo.au.tsinghua.edu.cn/member/xwwang

An Example Workflow by Using Rbowtie2

Installation

To install the latest version of Rbowtie2, you will need to be using the latest version of R. Rbowtie2 is part of Bioconductor project, so you can install Rbowtie2 and its dependencies like this:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("Rbowtie2")
```

Loading

Just like other R package, you need to load Rbowtie2 like this each time before using the package.

```
library(Rbowtie2)
```

AdapterRemoval

All package functions mentioned in this subsection use the shared library of AdapterRemoval.

Idetitify Adapter

If you know the adapter sequence of reads files, you can skip this step. Besides,single end data is not support for this function yet so adapter sequence has to be known .

reads_1 and reads_2 are raw paired-end reads file with fastq format. adapters is two adapters character vector.

```r
td <- tempdir()
reads_1 <- system.file(package="Rbowtie2", "extdata", "adrm", "reads_1.fq")
reads_2 <- system.file(package="Rbowtie2", "extdata", "adrm", "reads_2.fq")
adapters <- identify_adapters(file1=reads_1,file2=reads_2,basename=file.path(td,"reads"),"--threads 3",
```

```
##  [1] "AdapterRemoval"
##  [2] "--identify-adapters"
##  [3] "--file1"
##  [4] "C:/Users/WeiZheng/Documents/R/win-library/3.3/Rbowtie2/extdata/adrm/reads_1.fq"
##  [5] "--file2"
##  [6] "C:/Users/WeiZheng/Documents/R/win-library/3.3/Rbowtie2/extdata/adrm/reads_2.fq"
##  [7] "--threads"
##  [8] "3"
##  [9] "--basename"
## [10] "C:\\Users\\WeiZheng\\AppData\\Local\\Temp\\Rtmp0GrHNu/reads"
## Attempting to identify adapter sequences ...
##    Found 394 overlapping pairs ...
##    Of which 119 contained adapter sequence(s) ...
##
## Printing adapter sequences, including poly-A tails:
##   --adapter1:  AGATCGGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNATCTCGTATGCCGTCTTCTGCTTG
##                ||||||||||||||||||||||||||||||||||******||||||||||||||||||||||||
##    Consensus:  AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCACCTAATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAAAAAAAA
##      Quality:  55200522544444/4411330333330222222/1.1.1.1111100-00000///..+....--*-)),,++++++++**(('%
##
##     Top 5 most common 9-bp 5'-kmers:
##             1: AGATCGGAA = 96.00% (96)
##             2: AGAGCGAAA =  1.00% (1)
##             3: AGCTCGGAA =  1.00% (1)
##             4: AGATGGGAA =  1.00% (1)
##             5: AGATCGGGA =  1.00% (1)
##
##
##   --adapter2:  AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT
##                ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
##    Consensus:  AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATTAAAAAAAAAAAAAAAAAAAAAAAAAAAA
##      Quality:  525555555144141441430333303.2/22-2/-1..11111110--00000///..+....--*-),,,++++++++**(%'%
##
##     Top 5 most common 9-bp 5'-kmers:
##             1: AGATCGGAA = 100.00% (100)
```

```r
adapters
```

```
## [1] "AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCACCTAATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAAAAAAAAAA"
## [2] "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
```

Remove Adapter

With known adapter sequence, remove_adapter function can be call to trim adapters.

```r
remove_adapters(file1=reads_1,file2=reads_2,adapter1 = adapters[1], adapter2 = adapters[2],
output1=file.path(td,"reads_1.trimmed.fq"),output2=file.path(td,"reads_2.trimmed.fq"),
basename=file.path(td,"reads.base"),overwrite=TRUE,"--threads 3")
```

Additional Arguments and Version

If you need to set additional arguments like "–threads 3" above, you can call function below to print all options available. The fixed arguments like file1, file2 and basename etc. are invalid.

```
adapterremoval_usage()
```

You can get version information by call:

```
adapterremoval_version()
```

Bowtie2

All package functions mentioned in this subsection use the shared library of Bowtie2.

Build Bowtie2 Index

Before aligning reads, bowtie2 index should be build. refs is a character vector of fasta reference file paths. A prefix of bowtie index should be set to argument bt2Index. Then, 6 index files with .bt2 file name extension will be created with bt2Index prefix.

```
td <- tempdir()
refs <- dir(system.file(package="Rbowtie2", "extdata", "bt2","refs"),full=TRUE)
bowtie2_build(references=refs, bt2Index=file.path(td, "lambda_virus"),"--threads 4 --quiet",overwrite=T
```

Additional Arguments of Bowtie Build

If you need to set additional arguments like "–threads 4 –quiet" above, you can call function below to print all options available. The fixed arguments references, bt2Index are invalid.

```
bowtie2_build_usage()
```

```
## Bowtie 2 version 2.3.2 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
## Usage: bowtie2-build-s [options]* <reference_in> <bt2_index_base>
##     reference_in            comma-separated list of files with ref sequences
##     bt2_index_base          write bt2 data to files with this dir/basename
## *** Bowtie 2 indexes work only with v2 (not v1).  Likewise for v1 indexes. ***
## Options:
##     -f                      reference files are Fasta (default)
##     -c                      reference sequences given on cmd line (as
##                             <reference_in>)
##     -a/--noauto             disable automatic -p/--bmax/--dcv memory-fitting
##     -p/--packed             use packed strings internally; slower, less memory
##     --bmax <int>            max bucket sz for blockwise suffix-array builder
##     --bmaxdivn <int>        max bucket sz as divisor of ref len (default: 4)
##     --dcv <int>             diff-cover period for blockwise (default: 1024)
##     --nodc                  disable diff-cover (algorithm becomes quadratic)
##     -r/--noref              don't build .3/.4 index files
##     -3/--justref            just build .3/.4 index files
##     -o/--offrate <int>      SA is sampled every 2^<int> BWT chars (default: 5)
##     -t/--ftabchars <int>    # of chars consumed in initial lookup (default: 10)
##     --threads <int>         # of threads
##     --seed <int>            seed for random number generator
##     -q/--quiet              verbose output (for debugging)
##     -h/--help               print detailed description of tool and its options
##     --usage                   print this usage message
##     --version                 print version information and quit
```

Bowtie2 Alignment

The variable reads_1 and reads_1 are preprocessed reads file paths. With bowtie2 index, reads will be mapped to reference by calling bowtie2. The result is saved in a sam file whose path is set to samOutput

```
reads_1 <- system.file(package="Rbowtie2", "extdata", "bt2", "reads", "reads_1.fastq")
reads_2 <- system.file(package="Rbowtie2", "extdata", "bt2", "reads", "reads_2.fastq")
bowtie2(bt2Index = file.path(td, "lambda_virus"),samOutput = file.path(td, "result.sam"),
seq1=reads_1,seq2=reads_2,overwrite=TRUE,"--threads 3")
```

```
## 1000 reads; of these:
##   1000 (100.00%) were paired; of these:
##     82 (8.20%) aligned concordantly 0 times
##     918 (91.80%) aligned concordantly exactly 1 time
##     0 (0.00%) aligned concordantly >1 times
##     ----
##     82 pairs aligned concordantly 0 times; of these:
##       5 (6.10%) aligned discordantly 1 time
##     ----
##     77 pairs aligned 0 times concordantly or discordantly; of these:
##       154 mates make up the pairs; of these:
##         100 (64.94%) aligned 0 times
##         54 (35.06%) aligned exactly 1 time
##         0 (0.00%) aligned >1 times
## 95.00% overall alignment rate
```

```
head(readLines(file.path(td, "result.sam")))
```

```
## [1] "@HD\tVN:1.0\tSO:unsorted"
## [2] "@SQ\tSN:gi|9626243|ref|NC_001416.1|\tLN:48502"
## [3] "@PG\tID:bowtie2\tPN:bowtie2\tVN:2.3.2\tCL:\"bowtie2-align-s --threads 3 -x C:\\Users\\WeiZheng\
## [4] "r33\t99\tgi|9626243|ref|NC_001416.1|\t1304\t42\t119M\t=\t1500\t246\tNAAGCGTATTGAAGGCTCGGTCTGGCC
## [5] "r33\t147\tgi|9626243|ref|NC_001416.1|\t1500\t42\t50M\t=\t1304\t-246\tCCGGATGACCCCTCCAGCGTGTTTTA
## [6] "r17\t99\tgi|9626243|ref|NC_001416.1|\t29683\t42\t83M\t=\t29803\t232\tTTCNNNTAAANGCANTCAGCAACGNT
```

Additional Arguments and Version of Bowtie2 Aligner

If you need to set additional arguments like "–threads 3" above, you can call function below to print all options available. The fixed arguments like bt2Index, samOutput and seq1 etc. are invalid.

```
bowtie2_usage()
```

```
## Bowtie 2 version 2.3.2 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
## Usage:
##   bowtie2-align [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r> | --interleaved <i>} [-S <sam>]
##
##   <bt2-idx>  Index filename prefix (minus trailing .X.bt2).
##              NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
##   <m1>       Files with #1 mates, paired with files in <m2>.
##   <m2>       Files with #2 mates, paired with files in <m1>.
##   <r>        Files with unpaired reads.
##   <i>        Files with interleaved paired-end FASTQ reads
##   <sam>      File for SAM output (default: stdout)
```

```
##
##   <m1>, <m2>, <r> can be comma-separated lists (no whitespace) and can be
##   specified many times.  E.g. '-U file1.fq,file2.fq -U file3.fq'.
##
## Options (defaults in parentheses):
##
##  Input:
##   -q                 query input files are FASTQ .fq/.fastq (default)
##   --tab5             query input files are TAB5 .tab5
##   --tab6             query input files are TAB6 .tab6
##   --qseq             query input files are in Illumina's qseq format
##   -f                 query input files are (multi-)FASTA .fa/.mfa
##   -r                 query input files are raw one-sequence-per-line
##   -c                 <m1>, <m2>, <r> are sequences themselves, not files
##   -s/--skip <int>    skip the first <int> reads/pairs in the input (none)
##   -u/--upto <int>    stop after first <int> reads/pairs (no limit)
##   -5/--trim5 <int>   trim <int> bases from 5'/left end of reads (0)
##   -3/--trim3 <int>   trim <int> bases from 3'/right end of reads (0)
##   --phred33          qualities are Phred+33 (default)
##   --phred64          qualities are Phred+64
##   --int-quals        qualities encoded as space-delimited integers
##
##  Presets:                 Same as:
##   For --end-to-end:
##    --very-fast            -D 5 -R 1 -N 0 -L 22 -i S,0,2.50
##    --fast                 -D 10 -R 2 -N 0 -L 22 -i S,0,2.50
##    --sensitive            -D 15 -R 2 -N 0 -L 22 -i S,1,1.15 (default)
##    --very-sensitive       -D 20 -R 3 -N 0 -L 20 -i S,1,0.50
##
##   For --local:
##    --very-fast-local      -D 5 -R 1 -N 0 -L 25 -i S,1,2.00
##    --fast-local           -D 10 -R 2 -N 0 -L 22 -i S,1,1.75
##    --sensitive-local      -D 15 -R 2 -N 0 -L 20 -i S,1,0.75 (default)
##    --very-sensitive-local -D 20 -R 3 -N 0 -L 20 -i S,1,0.50
##
##  Alignment:
##   -N <int>           max # mismatches in seed alignment; can be 0 or 1 (0)
##   -L <int>           length of seed substrings; must be >3, <32 (22)
##   -i <func>          interval between seed substrings w/r/t read len (S,1,1.15)
##   --n-ceil <func>    func for max # non-A/C/G/Ts permitted in aln (L,0,0.15)
##   --dpad <int>       include <int> extra ref chars on sides of DP table (15)
##   --gbar <int>       disallow gaps within <int> nucs of read extremes (4)
##   --ignore-quals     treat all quality values as 30 on Phred scale (off)
##   --nofw             do not align forward (original) version of read (off)
##   --norc             do not align reverse-complement version of read (off)
##   --no-1mm-upfront   do not allow 1 mismatch alignments before attempting to
##                      scan for the optimal seeded alignments
##   --end-to-end       entire read must align; no clipping (on)
##    OR
##   --local            local alignment; ends might be soft clipped (off)
##
##  Scoring:
##   --ma <int>         match bonus (0 for --end-to-end, 2 for --local)
##   --mp <int>         max penalty for mismatch; lower qual = lower penalty (6)
##   --np <int>         penalty for non-A/C/G/Ts in read/ref (1)
##   --rdg <int>,<int>  read gap open, extend penalties (5,3)
##   --rfg <int>,<int>  reference gap open, extend penalties (5,3)
##   --score-min <func> min acceptable alignment score w/r/t read length
##                      (G,20,8 for local, L,-0.6,-0.6 for end-to-end)
##
##  Reporting:
##   (default)          look for multiple alignments, report best, with MAPQ
##    OR
##   -k <int>           report up to <int> alns per read; MAPQ not meaningful
##    OR
##   -a/--all           report all alignments; very slow, MAPQ not meaningful
##
##  Effort:
##   -D <int>           give up extending after <int> failed extends in a row (15)
##   -R <int>           for reads w/ repetitive seeds, try <int> sets of seeds (2)
##
##  Paired-end:
##   -I/--minins <int>  minimum fragment length (0)
##   -X/--maxins <int>  maximum fragment length (500)
##   --fr/--rf/--ff     -1, -2 mates align fw/rev, rev/fw, fw/fw (--fr)
##   --no-mixed         suppress unpaired alignments for paired reads
##   --no-discordant    suppress discordant alignments for paired reads
##   --dovetail         concordant when mates extend past each other
##   --no-contain       not concordant when one mate alignment contains other
##   --no-overlap       not concordant when mates overlap at all
##
##  Output:
##   -t/--time          print wall-clock time taken by search phases
##   --quiet            print nothing to stderr except serious errors
##   --met-file <path>  send metrics to file at <path> (off)
##   --met-stderr       send metrics to stderr (off)
##   --met <int>        report internal counters & metrics every <int> secs (1)
##   --no-unal          suppress SAM records for unaligned reads
##   --no-head          suppress header lines, i.e. lines starting with @
##   --no-sq            suppress @SQ header lines
##   --rg-id <text>     set read group id, reflected in @RG line and RG:Z: opt field
##   --rg <text>        add <text> ("lab:value") to @RG line of SAM header.
##                      Note: @RG line only printed when --rg-id is set.
##   --omit-sec-seq     put '*' in SEQ and QUAL fields for secondary alignments.
##   --sam-noqname-trunc Suppress standard behavior of truncating readname at first whitespace
##                       at the expense of generating non-standard SAM.
##
##  Performance:
##   -p/--threads <int> number of alignment threads to launch (1)
##   --reorder          force SAM output order to match order of input reads
##
##  Other:
##   --qc-filter        filter out reads that are bad according to QSEQ filter
##   --seed <int>       seed for random number generator (0)
##   --non-deterministic seed rand. gen. arbitrarily instead of using read attributes
##   --version          print version information and quit
##   -h/--help          print this usage message
```

You can get version information by call:

```
bowtie2_version()
```

```
## bowtie2-align-s version 2.3.2
## 64-bit
## Built on Rbowtie2
## 2017
## Compiler: C++11
## Options: -O3 -m64 -msse2 -funroll-loops -g3 -DPOPCNT_CAPABILITY
## Sizeof {int, long, long long, void*, size_t, off_t}: {4, 4, 8, 8, 8, 8}
```

Acknowledgement

We would like to thank Huan Fang for package testing and valuable suggestions.

References

[1] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature methods, 9(4), 357-359.

[2] Schubert, Lindgreen, and Orlando (2016). AdapterRemoval v2: rapid adapter trimming, identification, and read merging. BMC Research Notes, 12;9(1):88.