An Introduction to *Rbowtie*

Anita Lerch, Dimos Gaidatzis and Michael Stadler

Modified: November 27, 2012. Compiled: May 5, 2013

Contents

1 Introduction

The *Rbowtie* package provides an R wrapper around the popular bowtie[1] short read aligner and around SpliceMap[2] a de novo splice junction discovery and alignment tool, which makes use of the bowtie software package.

The package is used by the *QuasR*[3] bioconductor package to quantify and annotate short reads. 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.

2 Preliminaries

2.1 Citing *Rbowtie*

If you use *Rbowtie*[4] in your work, you can cite it as follows:

```
> citation("Rbowtie")
  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).

  Hahne F, Lerch A, Stadler MB. Rbowtie: An R wrapper for
  bowtie and SpliceMap short read aligners. (unpublished)

  Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast
  and memory-efficient alignment of short DNA sequences
  to the human genome. Genome Biology 10(3):R25 (2009).
```

2.2 Installation

Rbowtie is a package for the R computing environment and it is assumed that you have already installed R. See the R project at `http://www.r-project.org`. To install the latest version of *Rbowtie*, you will need to be using the latest version of R. *Rbowtie* is part of the Bioconductor project at `http://www.bioconductor.org`. To get *Rbowtie* together with its dependencies you can use

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

2.3 Loading of *Rbowtie*

In order to run the code examples in this vignette, the *Rbowtie* library need to be loaded.

```
> library(Rbowtie)
```

2.4 How to get help

Most questions about *Rbowtie* will hopefully be answered by the documentation or references. If you've run into a question which isn't addressed by the documentation, or you've found a conflict between the documentation and software itself, then there is an active support community which can offer help. The authors of the package (maintainer: maintainer("Rbowtie")) always appreciate receiving reports of bugs in the package functions or in the documentation.

The same goes for well-considered suggestions for improvements. Any other questions or problems concerning *Rbowtie* should be sent to the Bioconductor mailing list `bioconductor@stat.math.ethz.ch`. To subscribe to the mailing list, see `https://stat.ethz.ch/mailman/listinfo/bioconductor`. Please send requests for general assistance and advice

to the mailing list rather than to the individual authors. Users posting to the mailing list for the first time should read the helpful posting guide at `http://www.bioconductor.org/doc/postingGuide.html`. Note that each function in *Rbowtie* has it's own help page, e.g. `help("bowtie")`. Mailing list etiquette requires that you read the relevant help page carefully before posting a problem to the list.

3 Example usage for individual *Rbowtie* functions

Please refer to the *Rbowtie* reference manual or the function documentation (e.g. using `?bowtie`) for a complete description of *Rbowtie* functions. The descriptions provided below are meant to give and overview over all functions and summarize the purpose of each one.

3.1 Build the reference index with `bowtie_build`

To be able to align short reads to a genome, an index has to be build first using the function `bowtie_build`. Information about arguments can be found with the help of the `bowtie_build_usage` function or in the manual page `?bowtie_build`.

```
> bowtie_build_usage()
 [1] "Usage: bowtie-build [options]* <reference_in> <ebwt_outfile_base>"
 [2] "    reference_in            comma-separated list of files with ref sequences"
 [3] "    ebwt_outfile_base       write Ebwt data to files with this dir/basename"
 [4] "Options:"
 [5] "    -f                      reference files are Fasta (default)"
 [6] "    -c                      reference sequences given on cmd line (as <seq_in>)"
 [7] "    -C/--color              build a colorspace index"
 [8] "    -a/--noauto             disable automatic -p/--bmax/--dcv memory-fitting"
 [9] "    -p/--packed             use packed strings internally; slower, uses less mem"
[10] "    --bmax <int>            max bucket sz for blockwise suffix-array builder"
[11] "    --bmaxdivn <int>        max bucket sz as divisor of ref len (default: 4)"
[12] "    --dcv <int>             diff-cover period for blockwise (default: 1024)"
[13] "    --nodc                  disable diff-cover (algorithm becomes quadratic)"
[14] "    -r/--noref              don't build .3/.4.ebwt (packed reference) portion"
[15] "    -3/--justref            just build .3/.4.ebwt (packed reference) portion"
[16] "    -o/--offrate <int>      SA is sampled every 2^offRate BWT chars (default: 5)"
[17] "    -t/--ftabchars <int>    # of chars consumed in initial lookup (default: 10)"
[18] "    --ntoa                  convert Ns in reference to As"
[19] "    --seed <int>            seed for random number generator"
[20] "    -q/--quiet              verbose output (for debugging)"
[21] "    -h/--help               print detailed description of tool and its options"
[22] "    --usage                 print this usage message"
[23] "    --version               print version information and quit"
```

`refFiles` below is a vector with filenames of the reference sequence in `FASTA` format, and `indexDir` specifies an output directory for the index files that will be generated when calling `bowtie_build`:

```
> refFiles <- dir(system.file(package="Rbowtie", "samples", "refs"), full=TRUE)
> indexDir <- file.path(tempdir(), "refsIndex")
> tmp <- bowtie_build(references=refFiles, outdir=indexDir, prefix="index", force=TRUE)
> head(tmp)
[1] "Settings:"
[2] "  Output files: \"/tmp/RtmpPvvZrj/refsIndex/index.*.ebwt\""
[3] "  Line rate: 6 (line is 64 bytes)"
[4] "  Lines per side: 1 (side is 64 bytes)"
[5] "  Offset rate: 5 (one in 32)"
[6] "  FTable chars: 10"
```

3.2 Create alignment with `bowtie`

Information about the arguments supported by the `bowtie` function can be obtained with the help of the `bowtie_usage` function or in the manual page `?bowtie`.

```
> bowtie_usage()
 [1] "Usage: "
 [2] "  bowtie [options]* <ebwt> {-1 <m1> -2 <m2> | --12 <r> | <s>} [<hit>]"
 [3] ""
 [4] "  <m1>    Comma-separated list of files containing upstream mates (or the"
 [5] "          sequences themselves, if -c is set) paired with mates in <m2>"
 [6] "  <m2>    Comma-separated list of files containing downstream mates (or the"
 [7] "          sequences themselves if -c is set) paired with mates in <m1>"
 [8] "  <r>     Comma-separated list of files containing Crossbow-style reads.  Can be"
 [9] "          a mixture of paired and unpaired.  Specify \"-\" for stdin."
[10] "  <s>     Comma-separated list of files containing unpaired reads, or the"
[11] "          sequences themselves, if -c is set.  Specify \"-\" for stdin."
[12] "  <hit>   File to write hits to (default: stdout)"
[13] "Input:"
[14] "  -q                 query input files are FASTQ .fq/.fastq (default)"
[15] "  -f                 query input files are (multi-)FASTA .fa/.mfa"
[16] "  -r                 query input files are raw one-sequence-per-line"
[17] "  -c                 query sequences given on cmd line (as <mates>, <singles>)"
[18] "  -C                 reads and index are in colorspace"
[19] "  -Q/--quals <file>  QV file(s) corresponding to CSFASTA inputs; use with -f -C"
[20] "  --Q1/--Q2 <file>   same as -Q, but for mate files 1 and 2 respectively"
[21] "  -s/--skip <int>    skip the first <int> reads/pairs in the input"
[22] "  -u/--qupto <int>   stop after first <int> reads/pairs (excl. skipped reads)"
[23] "  -5/--trim5 <int>   trim <int> bases from 5' (left) end of reads"
[24] "  -3/--trim3 <int>   trim <int> bases from 3' (right) end of reads"
[25] "  --phred33-quals    input quals are Phred+33 (default)"
[26] "  --phred64-quals    input quals are Phred+64 (same as --solexa1.3-quals)"
[27] "  --solexa-quals     input quals are from GA Pipeline ver. < 1.3"
[28] "  --solexa1.3-quals  input quals are from GA Pipeline ver. >= 1.3"
[29] "  --integer-quals    qualities are given as space-separated integers (not ASCII)"
[30] "Alignment:"
[31] "  -v <int>           report end-to-end hits w/ <=v mismatches; ignore qualities"
[32] "    or"
[33] "  -n/--seedmms <int> max mismatches in seed (can be 0-3, default: -n 2)"
[34] "  -e/--maqerr <int>  max sum of mismatch quals across alignment for -n (def: 70)"
[35] "  -l/--seedlen <int> seed length for -n (default: 28)"
[36] "  --nomaqround       disable Maq-like quality rounding for -n (nearest 10 <= 30)"
[37] "  -I/--minins <int>  minimum insert size for paired-end alignment (default: 0)"
[38] "  -X/--maxins <int>  maximum insert size for paired-end alignment (default: 250)"
[39] "  --fr/--rf/--ff     -1, -2 mates align fw/rev, rev/fw, fw/fw (default: --fr)"
[40] "  --nofw/--norc      do not align to forward/reverse-complement reference strand"
[41] "  --maxbts <int>     max # backtracks for -n 2/3 (default: 125, 800 for --best)"
[42] "  --pairtries <int>  max # attempts to find mate for anchor hit (default: 100)"
[43] "  -y/--tryhard       try hard to find valid alignments, at the expense of speed"
[44] "  --chunkmbs <int>   max megabytes of RAM for best-first search frames (def: 64)"
[45] "Reporting:"
[46] "  -k <int>           report up to <int> good alignments per read (default: 1)"
[47] "  -a/--all           report all alignments per read (much slower than low -k)"
[48] "  -m <int>           suppress all alignments if > <int> exist (def: no limit)"
[49] "  -M <int>           like -m, but reports 1 random hit (MAPQ=0); requires --best"
```

```
 [50] "  --best             hits guaranteed best stratum; ties broken by quality"
 [51] "  --strata           hits in sub-optimal strata aren't reported (requires --best)"
 [52] "Output:"
 [53] "  -t/--time          print wall-clock time taken by search phases"
 [54] "  -B/--offbase <int> leftmost ref offset = <int> in bowtie output (default: 0)"
 [55] "  --quiet            print nothing but the alignments"
 [56] "  --refout           write alignments to files refXXXXX.map, 1 map per reference"
 [57] "  --refidx           refer to ref. seqs by 0-based index rather than name"
 [58] "  --al <fname>       write aligned reads/pairs to file(s) <fname>"
 [59] "  --un <fname>       write unaligned reads/pairs to file(s) <fname>"
 [60] "  --max <fname>      write reads/pairs over -m limit to file(s) <fname>"
 [61] "  --suppress <cols>  suppresses given columns (comma-delim'ed) in default output"
 [62] "  --fullref          write entire ref name (default: only up to 1st space)"
 [63] "Colorspace:"
 [64] "  --snpphred <int>   Phred penalty for SNP when decoding colorspace (def: 30)"
 [65] "     or"
 [66] "  --snpfrac <dec>    approx. fraction of SNP bases (e.g. 0.001); sets --snpphred"
 [67] "  --col-cseq         print aligned colorspace seqs as colors, not decoded bases"
 [68] "  --col-cqual        print original colorspace quals, not decoded quals"
 [69] "  --col-keepends     keep nucleotides at extreme ends of decoded alignment"
 [70] "SAM:"
 [71] "  -S/--sam           write hits in SAM format"
 [72] "  --mapq <int>       default mapping quality (MAPQ) to print for SAM alignments"
 [73] "  --sam-nohead       supppress header lines (starting with @) for SAM output"
 [74] "  --sam-nosq         supppress @SQ header lines for SAM output"
 [75] "  --sam-RG <text>    add <text> (usually \"lab=value\") to @RG line of SAM header"
 [76] "Performance:"
 [77] "  -o/--offrate <int> override offrate of index; must be >= index's offrate"
 [78] "  -p/--threads <int> number of alignment threads to launch (default: 1)"
 [79] "  --mm               use memory-mapped I/O for index; many 'bowtie's can share"
 [80] "  --shmem            use shared mem for index; many 'bowtie's can share"
 [81] "Other:"
 [82] "  --seed <int>       seed for random number generator"
 [83] "  --verbose          verbose output (for debugging)"
 [84] "  --version          print version information and quit"
 [85] "  -h/--help          print this usage message"
```

In the example below, `readsFiles` is the name of a file containing short reads to be aligned with `bowtie`, and `samFiles` specifies the name of the output file with the generated alignments.

```
> readsFiles <- system.file(package="Rbowtie", "samples", "reads", "reads.fastq")
> samFiles <- file.path(tempdir(), "alignments.sam")
> bowtie(sequences=readsFiles,
+        index=file.path(indexDir, "index"),
+        outfile=samFiles, sam=TRUE,
+        best=TRUE, force=TRUE)
> strtrim(readLines(samFiles), 65)
[1] "@HD\tVN:1.0\tSO:unsorted"
[2] "@SQ\tSN:chr1\tLN:100000"
[3] "@SQ\tSN:chr2\tLN:100000"
[4] "@SQ\tSN:chr3\tLN:100000"
[5] "@PG\tID:Bowtie\tVN:1.0.0\tCL:\"/tmp/Rtmpp1GLdT/Rinst52855d0a9b96/Rbowtie"
[6] "HWUSI-EAS1513_0012:6:48:5769:946#0/1\t0\tchr1\t819\t255\t101M\t*\t0\t0\tTGGAGTTCATG"
[7] "HWUSI-EAS1513_0012:6:48:6908:952#0/1\t0\tchr2\t1133\t255\t101M\t*\t0\t0\tAACATAGTGA"
```

```
[8] "HWUSI-EAS1513_0012:6:48:8070:953#0/1\t0\tchr1\t7543\t255\t101M\t*\t0\t0\tGTCTGTCTAG"
[9] "HWUSI-EAS1513_0012:6:48:9942:949#0/1\t4\t*\t0\t0\t*\t*\t0\t0\tCGGTTCCTGTATCCTTAATAA"
```

3.3 Create spliced alignment with SpliceMap

While bowtie only generates ungapped alignments, the SpliceMap function can be used to generate spliced alignments. SpliceMap is itself using bowtie. To use it, it is necessary to create an index of the reference sequence as described in 3.1. SpliceMap parameters are specified in the form of a named list, which follows closely the configure file format of the original SpliceMap program[2]. Be aware that SpliceMap can only be used for reads that are at least 50bp long.

```
> readsFiles <- system.file(package="Rbowtie", "samples", "reads", "reads.fastq")
> refDir <- system.file(package="Rbowtie", "samples", "refs", "chr1.fa")
> indexDir <- file.path(tempdir(), "refsIndex")
> 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)
> res
[1] "/tmp/RtmpPvvZrj/splicedAlignments.sam"
> strtrim(readLines(samFiles), 65)
[1] "@HD\tVN:1.0\tSO:coordinate"
[2] "@SQ\tSN:chr1\tLN:100000"
[3] "@PG\tID:SpliceMap\tVN:3.3.5.2 (55)"
[4] "HWUSI-EAS1513_0012:6:48:5769:946#0\t0\tchr1\t819\t255\t101M\t*\t0\t0\tTGGAGTTCATGTG"
[5] "HWUSI-EAS1513_0012:6:48:6908:952#0\t4\t*\t0\t0\t*\t*\t0\t0\tAACATAGTGAAGAAACCTCATAG"
[6] "HWUSI-EAS1513_0012:6:48:8070:953#0\t0\tchr1\t7543\t255\t101M\t*\t0\t0\tGTCTGTCTAGTG"
[7] "HWUSI-EAS1513_0012:6:48:9942:949#0\t4\t*\t0\t0\t*\t*\t0\t0\tCGGTTCCTGTATCCTTAATAAGT"
```

4 Session information

The output in this vignette was produced under:

```
> sessionInfo()
R version 3.0.0 (2013-04-03)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
```

```
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=C                 LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets
[7] methods   base

other attached packages:
[1] Rbowtie_1.0.3

loaded via a namespace (and not attached):
[1] tools_3.0.0
```

References

[1] B. Langmead, C. Trapnell, M. Pop, and S.L. Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

[2] K.F. Au, H. Jiang, L. Lin, Y. Xing, and W.H. Wong. Detection of splice junctions from paired-end rna-seq data by splicemap. *Nucleic Acids Research*, 38(14):4570–4578, 2010.

[3] A. Lerch, D. Gaidatzis, F. Hahne, and M.B. Stadler. Quasr: Quantify and annotate short reads in r. unpublished, 2012.

[4] F. Hahne, A. Lerch, and M.B. Stadler. bowtie: An r wrapper for bowtie and splicemap short read aligners. unpublished, 2012.