BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 23 2014, pages 3421–3423
doi:10.1093/bioinformatics/btu533

Gene expression

Advance Access publication August 13, 2014

READemption—a tool for the computational analysis of deep-sequencing–based transcriptome data

Konrad U. Förstner[1,2], Jörg Vogel[1] and Cynthia M. Sharma[2,*]

[1]Institute for Molecular Infection Biology and [2]Research Centre for Infectious Diseases (ZINF), University of Würzburg, D-97080 Würzburg, Germany

Associate Editor: Ivo Hofacker

ABSTRACT

Summary: RNA-Seq has become a potent and widely used method to qualitatively and quantitatively study transcriptomes. To draw biological conclusions based on RNA-Seq data, several steps, some of which are computationally intensive, have to be taken. Our READemption pipeline takes care of these individual tasks and integrates them into an easy-to-use tool with a command line interface. To leverage the full power of modern computers, most subcommands of READemption offer parallel data processing. While READemption was mainly developed for the analysis of bacterial primary transcriptomes, we have successfully applied it to analyze RNA-Seq reads from other sample types, including whole transcriptomes and RNA immunoprecipitated with proteins, not only from bacteria but also from eukaryotes and archaea.

Availability and implementation: READemption is implemented in Python and is published under the ISC open source license. The tool and documentation is hosted at http://pythonhosted.org/READemption (DOI:10.6084/m9.figshare.977849).

Contact: cynthia.sharma@uni-wuerzburg.de and konrad.foerstner@uni-wuerzburg.de

Received on April 1, 2014; revised on July 13, 2014; accepted on August 1, 2014

1 INTRODUCTION

RNA-Seq, the examination of cDNA by massively parallel sequencing technologies, is a potent way to perform transcriptome analyses at single-nucleotide resolution and with a high dynamic range (Wang *et al*., 2009). It has been successfully used to annotate transcript boundaries and to identify novel transcripts such as small regulatory RNAs in both prokaryotes and eukaryotes (Filiatrault, 2011; Ozsolak and Milos, 2011). Most prominently, it can be applied to quantify the expression levels of genes, having been shown to be more powerful to detect changes in gene expression than microarrays (Zhao *et al*., 2014). It can also be used to study the interaction of proteins and RNAs through performing RNA-Seq of coimmunoprecipation (coIP) samples (König *et al*., 2012). Likewise, any other subset of RNA molecules from, for instance, RNA size-fractionation, ribosome profiling, metatranscriptomics or degradome profiling experiments can be sequenced. Owing to decreasing costs and ever increasing speed of deep sequencing, the bioinformatical analysis has become a bottleneck of RNA-Seq–based projects.

*To whom correspondence should be addressed.

We have created an automated RNA-Seq processing pipeline named READemption with the initial purpose to handle differential RNA-Seq (dRNA-Seq) data for the determination of transcriptional start sites in bacteria (Sharma *et al*., 2010, Sharma and Vogel, 2014). We saw the need for this, as other available RNA-Seq analysis pipelines (e.g. Delhomme *et al*., 2012, McClure *et al*., 2013) were not designed for this application. Additionally, while most available RNA-Seq pipelines put priority on fast mapping, we have chosen `segemehl` as short read aligner (Hoffmann *et al*., 2009). This mapper has a relatively high demand of memory and computation capacities but, in return, it offers high sensitivity as well as a low false-discovery rate and can perform multiple splitting of reads (Otto *et al*., 2014). We have since extended the functionality of this Python-based pipeline, so that it is now capable of analyzing RNA-Seq reads from a wide range of experiments. We have successfully applied READemption for the analyses of RNA samples from bacterial, archaeal and eukaryotic species as well as for RNA virus genomes (e.g. Dugar *et al*., 2013; Zhelyazkova *et al*., 2012). It is able to work with reads from both Illumina and 454 platforms of different lengths and can be used for single- and paired-end sequenced libraries.

2 DESCRIPTION

READemption integrates the steps that are required to interpret and gain biological knowledge from RNA-Seq experiments in one tool and makes them accessible via a consistent command line interface. Additionally, it conducts parallel data processing to reduce the runtime. The tool performs quality trimming, poly(A) and adapter clipping as well as size filtering of raw cDNA reads from different sequencing platforms, mapping to reference sequences, coverage calculation, gene-based quantification and comparison of expression levels. A summary of the pipeline's workflow is depicted in the flow chart in Figure 1A. Moreover, it provides several statistics such as read mappability and generates plots and files for the visualization of the results in genome browsers (for examples, see Fig. 1B).

READemption was designed as high-performance application and follows the concept of 'convention over configuration'. This includes the use of established default parameter values and the approach that files are placed or linked into defined paths and are then treated accordingly. The names of the input read files are used to generate names for the associated output files. Though the described design principle, READemption offers

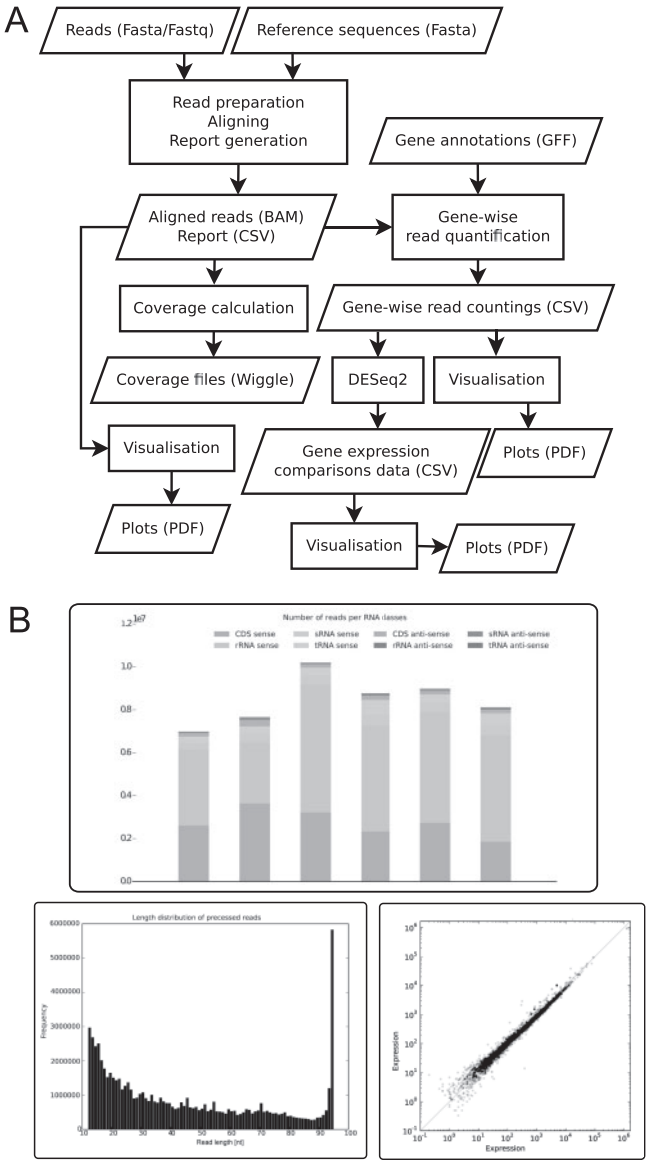

Fig. 1. (**A**) Data and work flow of READemption including the input, output and the performed steps. (**B**) Examples of plots generated by READemption (a bar plot of the read distribution mapped to the different RNA classes; a scatterplot of gene expression levels of two libraries; a histogram of read length after poly(A) trimming)

several parameters, which enable the user to adapt its execution to the specific needs.

READemption provides the subcommands `align`, `coverage`, `gene_quanti`, `deseq`, `viz_align`, `viz_gene_quanti` and `viz_deseq`, which combine several processing substeps into comprehensible units.

Read processing and mapping: The fundamental tasks of preprocessing the input reads and aligning them to reference sequences is covered by the subcommand `align`. In an initial step, READemption parses the input read files in Fasta or Fastq format, performs quality trimming, removes adapters and/or poly(A)-tails introduced during the library preparation and discards reads shorter than a given cutoff (default 12 nt). For the alignment of reads to reference sequences, the short read mapper `segemehl` and its remapper `lack` (Hoffmann *et al*., 2009, Otto *et al*., 2014) are used. To have high confidence of read mappings even for short sequences, the required mapping accuracy of 95% is used per default. The mapping is followed by the conversion of the resulting Sequence Alignment/Map format (SAM) alignment files into Binary Alignment/Map format (BAM) files and the generation of mapping statistics. The latter summarizes the numbers of uniquely and multiple mapped reads as well as the number of alignments, clipping and filtering events for each genomic element in table format.

Coverage calculation: Based on the read alignments provided in the BAM files, cDNA coverage files can be generated using the subcommand `coverage`. It creates several wiggle files that are based on different normalization methods such as the division by the total number of aligned reads and represents the nucleotide-wise cDNA coverage in a strand-specific manner. To visually inspect the reads mapped to the individual cDNA libraries and to compare them among each other, these wiggle files can be loaded into common genome browsers.

Gene expression quantification: The read alignments can also be further used by the subcommand `gene_quanti` to calculate gene-wise read counts. For this purpose, annotation files including gene positions in GFF3 format (Gene feature format) have to be provided. For each annotation entry, the number of reads that are overlapping with a user-defined number of nucleotides (default 1 nt) are reported. To also detect non-annotated antisense transcripts, the reads which are mapped in antisense direction to a given annotation can be quantified. Besides raw gene-wise read countings, normalized values—by total number of aligned reads as well as reads per kilobase per exon model per million mapped reads (RPKM) (Mortazavi *et al*., 2008)—are returned.

Differential gene expression analysis: For pairwise expression comparison, the subcommand `deseq` offers statistical analysis based on the approach implemented in `DESeq2` (Anders and Huber, 2010, Love *et al*., 2014), which builds on the raw read counting and is a widely adapted and intensively tested library (Guo *et al*., 2013, Rapaport *et al*., 2013, Reeb and Steibel, 2013). The user has to specify the conditions of the libraries to let DESeq2 treat replicates accordingly. The results of `DESeq2` are reformated and supplemented with the provided gene annotations.

Plotting: The final three subcommands called `viz_align`, `viz_gene_quanti` and `viz_deseq` generate several visualizations that help to interpret the result of the subcommand `align`, `gene_quanti` and `deseq`, respectively. The diverse plots contain among others histograms of the read length distributions before and after the read clipping, volcano plots and MA plots (log ratios (M) versus arithmetic mean of expression values (A)) of the differential gene expression analysis.

READemption requires `Python` 3.2 or higher (http://python.org), `Biopython` (Cock, 2009), `matplotlib` (Hunter, 2007) and `numpy` (Oliphant, 2007) as well as the `samtools` (Li *et al*., 2009) wrapper `pysam` (http://pypi.python.org/pypi/pysam/). The subcommand `deseq` relies on an `R` (http://cran.r-project.org) installation and the bioconductor package `DESeq2` (Anders and Huber, 2010, Love *et al*., 2014). Instructions how to install READemption and how to execute its subcommands including

examples can be found in the documentation. Additionally, an Ubuntu live and installation image with READemption preinstalled is available for download.

3 CONCLUSIONS

We present an open source pipeline for the analysis of RNA-Seq data from all domains of life. READemption generates several output files that can be examined with common office suites, graphic programs and genome browsers. Its features make it a useful tool for anybody interested in the computational analysis of RNA-Seq data with the required basic command line skills.

ACKNOWLEDGEMENTS

The authors thank members of the Sharma and the Vogel groups, especially Thorsten Bischler and Lei Li for testing and constructive feedback.

Funding: Work in the Sharma and Vogel laboratories is supported by the Bavarian Research Network for Molecular Biosystems (BioSysNet). The JV laboratory received financial support from a BMBF eBio grant RNAsys and DFG project VO 875/4-2. The CMS laboratory received financial support from the ZINF Young Investigator program at the Research Center for Infectious Diseases (ZINF, Würzburg, Germany), DFG project Sh580/1-1, and the Young Academy program of the Bavarian Academy of Sciences.

Conflict of interest: none declared.

REFERENCES

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R10.

Cock,P.J. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Delhomme,N. *et al.* (2012) easyRNASeq: a bioconductor package for processing RNA-Seq data. *Bioinformatics*, **28**, 2532–2533.

Dugar,G. *et al.* (2013) High-resolution transcriptome maps reveal strain-specific regulatory features of multiple *Campylobacter jejuni* isolates. *PLoS Genet.*, **9**, e1003495.

Filiatrault,M.J. (2011) Progress in prokaryotic transcriptomics. *Curr. Opin. Microbiol.*, **14**, 579–586.

Guo,Y. *et al.* (2013) Evaluation of read count based RNAseq analysis methods. *BMC Genomics*, **14 (Suppl. 8)**, S2.

Hoffmann,S. *et al.* (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput. Biol.*, **5**, e1000502.

Hunter,J.D. (2007) Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90.

König,J. *et al.* (2012) Protein-RNA interactions: new genomic technologies and perspectives. *Nat. Rev. Genet.*, **13**, 77–83.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Love,M. *et al.* (2014) Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *bioRxiv*, preprint doi:10.1101/002832.

McClure,R. *et al.* (2013) Computational analysis of bacterial RNA-Seq data. *Nucleic Acids Res.*, **41**, e140.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, **5**, 621–628.

Oliphant,T.E. (2007) Python for scientific computing. *Comput. Sci. Eng.*, **9**, 90.

Otto,C. *et al.* (2014) Lacking alignments? The next generation sequencing mapper segemehl revisited. *Bioinformatics*, **30**, 1837–1843.

Ozsolak,F. and Milos,P.M. (2011) RNA sequencing: advances, challenges and opportunities. *Nat. Rev. Genet.*, **12**, 87–98.

Rapaport,F. *et al.* (2013) Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biol.*, **14**, R95.

Reeb,P.D. and Steibel,J.P. (2013) Evaluating statistical analysis models for RNA sequencing experiments. *Front Genet.*, **4**, 178.

Sharma,C.M. *et al.* (2010) The primary transcriptome of the major human pathogen *Helicobacter pylori*. *Nature*, **2**, 14.

Sharma,C.M. and Vogel,J. (2014) Differential RNA-seq: the approach behind and the biological insight gained. *Curr. Opin. Microbiol.*, **19**, 97–105.

Wang,Z. *et al.* (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.

Zhao,S. *et al.* (2014) Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLoS One*, **9**, e78644.

Zhelyazkova,P. *et al.* (2012) The primary transcriptome of barley chloroplasts: numerous noncoding RNAs and the dominating role of the plastid-encoded RNA polymerase. *Plant Cell*, **24**, 123–136.