BIOINFORMATICS APPLICATIONS NOTE

Vol. 28 no. 16 2012, pages 2184–2185
doi:10.1093/bioinformatics/bts356

Sequence analysis

Advance Access publication June 27, 2012

RSeQC: quality control of RNA-seq experiments

Liguo Wang[1,2], Shengqin Wang[3] and Wei Li[1,2,*]

[1]Division of Biostatistics, Dan L. Duncan Cancer Center and [2]Department of Molecular and Cellular Biology, Baylor College of Medicine, Houston, TX 77030, USA and [3]State Key Laboratory of Bioelectronics, School of Biological Science and Medical Engineering, Southeast University, Nanjing, China

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: RNA-seq has been extensively used for transcriptome study. Quality control (QC) is critical to ensure that RNA-seq data are of high quality and suitable for subsequent analyses. However, QC is a time-consuming and complex task, due to the massive size and versatile nature of RNA-seq data. Therefore, a convenient and comprehensive QC tool to assess RNA-seq quality is sorely needed.

Results: We developed the RSeQC package to comprehensively evaluate different aspects of RNA-seq experiments, such as sequence quality, GC bias, polymerase chain reaction bias, nucleotide composition bias, sequencing depth, strand specificity, coverage uniformity and read distribution over the genome structure. RSeQC takes both SAM and BAM files as input, which can be produced by most RNA-seq mapping tools as well as BED files, which are widely used for gene models. Most modules in RSeQC take advantage of R scripts for visualization, and they are notably efficient in dealing with large BAM/SAM files containing hundreds of millions of alignments.

Availability and implementation: RSeQC is written in Python and C. Source code and a comprehensive user's manual are freely available at: http://code.google.com/p/rseqc/.

Contact: WL1@bcm.edu

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on February 27, 2012; revised on June 8, 2012; accepted on June 18, 2012

1 INTRODUCTION

Deep transcriptome sequencing (RNA-seq) provides massive and valuable information concerning all transcribed elements in the genome. Using RNA-seq, researchers are able to, for instance, profile gene expression, interrogate alternative splicing, identify novel transcripts and detect aberrant transcripts and coding variants. RNA-seq experiments should ideally be able to directly identify and quantify all RNA species, regardless of their size or frequency. However, current RNA-seq protocols still possess several intrinsic biases and limitations, such as nucleotide composition bias, GC bias and PCR bias. These biases directly affect the accuracy of many RNA-seq applications (Benjamini and Speed, 2012; Hansen and Brenner, 2010) and can be directly checked from raw sequences using tools like FastQC. However, these raw sequence-based metrics are not sufficient to ensure the usability of RNA-seq data; other RNA-seq-specific quality control (QC) metrics, such as sequencing depth, read distribution and coverage uniformity, are even more important. For instance, sequencing depth must be saturated before carrying out many RNA-seq applications, including expression profiling, alternative splicing analysis, novel isoform identification and transcriptome reconstruction. The use of RNA-seq with unsaturated sequencing depth gives imprecise estimations (such as for RPKM and splicing index) and fails to detect low abundance splice junctions, thereby limit the precision of many analyses. At the same time, sequencing depth is directly related to the cost of analysis. For an RNA-seq dataset close to saturation, additional sequencing is not cost-effective, as it would provide little additional information. Currently, a few tools are available for the QC of high-throughput sequencing data, but most of them (FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/), htSeqTools, FASTX-ToolKit (http://hannonlab.cshl.edu/fastx_toolkit/) and SAMStat) only focus on raw sequence-related metrics (Goecks *et al.*, 2010; Lassmann *et al.*, 2011; Planet *et al.*, 2012; Reich *et al.*, 2006). RNA-SeQC is the only tool designed for RNA-seq QC, but it still lacks many important functions, such as saturation checking (Deluca *et al.*, 2012). To address these needs, we have developed RSeQC to comprehensively assess the quality of RNA-seq experiments performed on clinical samples or other well-annotated model organisms, such as *mouse*, *fly*, *Caenorhabditis elegans* and yeast. RSeQC contains basic modules to evaluate raw sequence quality, RNA-seq-specific modules to perform annotation-based checking and utility modules for data visualization (Supplementary Fig. S1). Comparison with other QC tools indicates not only that RSeQC is more comprehensive and efficient but also that it has several unique checks not available elsewhere (Supplementary Table S1).

2 FEATURES AND METHODS

RSeQC consists of a series of Python programs to evaluate RNA-seq experiments from different aspects. Below are some selected modules from RSeQC:

(1) 'bam_stat.py' is used to check the mapping statistics of reads that are QC failed, unique mapped, splice mapped, mapped in proper pair, etc.

(2) 'inner_distance.py' is used to estimate the inner distance distribution between paired reads. The estimated inner distance should be consistent with gel size selection. This is an important parameter when using RNA-seq data to detect structure variation or aberrant splicing.

(3) 'geneBody_coverage.py' scales all transcripts to 100 nt and calculates the number of reads covering each nucleotide position. Finally, it generates a plot illustrating the coverage profile along the gene body (Fig. 1A).

[*]To whom correspondence should be addressed.

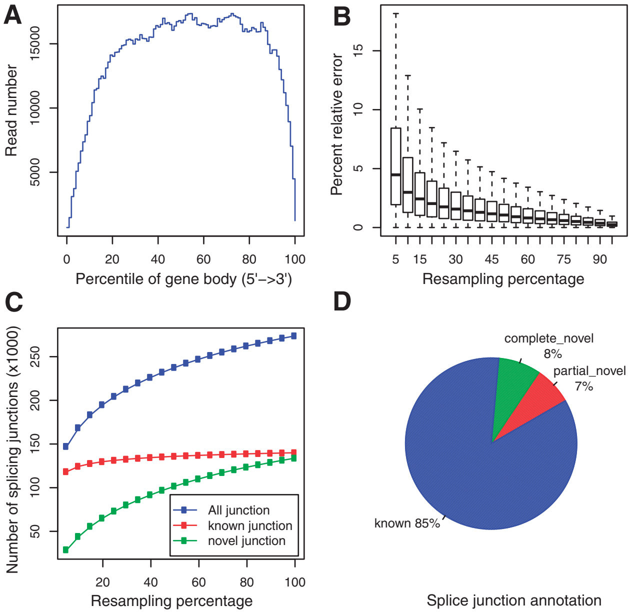

Fig. 1. Examples of RSeQC output. **(A)** Coverage uniformity over gene body. All transcripts were scaled into 100 nt. **(B)** Saturation analysis of expression for 25% highest expressed genes. **(C)** Saturation analysis of junction detection. **(D)** Annotation of detected splice junctions. ‘known’: splice junctions with both 5′ splice site (5′SS) and 3′ splice site (3′SS) annotated by reference gene model; ‘complete novel’: splice junctions with neither 5′SS nor 3′SS annotated by reference gene model; ‘partial novel’: splice junctions with either 5′SS or 3′SS annotated by reference gene model

(4) ‘read_distribution.py’ calculates the fraction of reads mapped to coding exons, 5′-untranslated region (UTR) exons, 3′-UTR exons, introns and intergenic regions based on the gene model provided. This module roughly reflects the uniformity of coverage; for example, reads are generally over-represented in 3′-UTR for the polyA + RNA-seq protocol. One can also apply this module to estimate the background noise level.

(5) ‘RPKM_saturation.py’ determines the precision of estimated RPKMs at the current sequencing depth by resampling (jackknifing) the total mapped reads. We use percent relative error ($100\times|\mathrm{RPKM}_{\mathrm{obs}}-\mathrm{RPKM}_{\mathrm{real}}|/\mathrm{RPKM}_{\mathrm{real}}$) to measure the precision of estimated RPKM (Fig. 1B). In practice, it is impossible to evaluate $\mathrm{RPKM}_{\mathrm{real}}$, and we use RPKM estimated from total reads to approximate $\mathrm{RPKM}_{\mathrm{real}}$.

(6) ‘junction_saturation.py’ determines if the current sequencing depth is sufficient to perform alternative splicing analyses. The concept is similar to that of ‘RPKM_saturation.py’: splice junctions are detected for each re-sampled subset of reads, and the number of detected splice junctions will increase as the resample percentage increases before finally reaching a fixed value. The junction saturation test is very important for alternative splicing analysis, as using an unsaturated sequencing depth would miss many rare splice junctions. (Fig. 1C).

(7) ‘infer_experiment.py’ is used to speculate the experimental design by sampling a subset of reads from the BAM file and comparing their genome coordinates and strands with those of the reference gene model. This module can determine if the given RNA-seq has been sequenced with paired-end or single-end reads. The module can also gauge whether sequencing is strand-specific, and if so, how reads are stranded.

(8) ‘junction_annotation.py’ separates all detected splice junctions into ‘known’, ‘complete novel’ and ‘partial novel’ by comparing them with the reference gene model (Fig. 1D).

(9) ‘RPKM_count.py’ calculates the raw read count and RPKM values for each exon, intron and mRNA region defined by the reference gene model.

(10) ‘bam2wig.py’ can efficiently convert a BAM file into a wiggle file for visualization. Wiggle files can be easily converted to bigwig files using the UCSC wigToBigWig tool.

3 RESULTS AND CONCLUSIONS

In summary, the RSeQC package provides a number of useful modules that can comprehensively evaluate RNA-seq data. ‘Basic modules’ quickly inspect sequence quality, nucleotide composition bias, PCR bias and GC bias, whereas ‘RNA-seq specific modules’ investigate the sequencing saturation status of both splice junction detection and expression estimation. These modules also inspect the mapped read-clipping profile, mapped read distribution, coverage uniformity over the gene body, reproducibility, strand specificity and splice junction annotation. Finally, RSeQC includes several useful tools to manipulate and normalize BigWig files for data visualization.

Funding: The Department of Defense Prostate Cancer Program PC094421 and the Cancer Prevention and Research Institute of Texas (RP110471-C3).

Conflict of Interest: none declared.

REFERENCES

Benjamini,Y. and Speed,T.P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.*, **40**, e72.

Deluca,D.S. *et al.* (2012) RNA-SeQC: RNA-seq metrics for quality control and process optimization. *Bioinformatics (Oxford, England)*, **28**, 1530–1532.

Goecks,J. *et al.* (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, **11**, R86.

Hansen,K. and Brenner,S. (2010) Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Res.*, **38**, e131.

Lassmann,T. *et al.* (2011) SAMStat: monitoring biases in next generation sequencing data. *Bioinformatics (Oxford, England)*, **27**, 130–131.

Planet,E. *et al.* (2012) htSeqTools: high-throughput sequencing quality control, processing and visualization in R. *Bioinformatics (Oxford, England)*, **28**, 589–590.

Reich,M. *et al.* (2006) GenePattern 2.0. *Nat. Genet.*, **38**, 500–501.