Gene expression

Rcount: simple and flexible RNA-Seq read counting
Marc W. Schmid* and Ueli Grossniklaus
Institute of Plant Biology and Zürich-Basel Plant Science Center, University of Zurich, 8008 Zürich, Switzerland

ABSTRACT
Summary: Analysis of differential gene expression by RNA sequencing (RNA-Seq) is frequently done using feature counts, i.e., the number of reads mapping to a gene. However, commonly used count algorithms (e.g. HTSeq) do not address the problem of reads aligning with multiple locations in the genome (multireads) or reads aligning with positions where two or more genes overlap (ambiguous reads). Rcount specifically addresses these issues. Furthermore, Rcount allows the user to assign priorities to certain feature types (e.g. higher priority for protein-coding genes compared to rRNA-coding genes) or to add flanking regions.

Availability and implementation: Rcount provides a fast and easy-to-use graphical user interface requiring no command line or programming skills. It is implemented in C++ using the SeqAn (www.sean.de) and the Qt libraries (qt-project.org). Source code and 64 bit binaries for (Ubuntu) Linux, Windows (7) and MacOSX are released under the GPLv3 license and are freely available on github.com/MWSchmid/Rcount.

Contact: marcschmid@gmx.ch
Supplementary information: Test data, genome annotation files, useful Python and R scripts and a step-by-step user guide (including run-time and memory usage tests) are available on github.com/MWSchmid/Rcount.

Received on July 15, 2014; revised on September 24, 2014; accepted on October 13, 2014

1 INTRODUCTION
RNA-Seq is the method of choice for transcriptional profiling and differential expression (DE) studies. For DE analysis, methods based on negative binomial modeling, such as the popular DESeq (Anders et al., 2010) and edgeR (Robinson et al., 2010), have been shown to outperform other methods in terms of specificity, sensitivity and control of false positives (Rapaport et al., 2013). Current work flows for DE analysis generally involve the (i) alignment of the short reads to a reference genome, (ii) quantification of expression levels and (iii) comparison between different treatments, tissue/cell types and time-points (Anders et al., 2013).

Read counting and read summarization are essential steps in any RNA-Seq workflow. However, they have received little attention. Specifically for RNA-Seq, counting is not as simple as it may appear. First, a read may align multiple times with the genome (multireads). Second, several genes may overlap at a given position within the genome. Reads aligning with those positions are ambiguous with respect to the gene they originate from (ambiguous reads). Third, alignments can span exon-junctions (exon-junction reads). Furthermore, a gene may have several isoforms. However, DE analysis is often performed using the total number of reads per gene.

Popular open source tools for read counting, such as HTSeq (www-huber.embl.de/users/anders/HTSeq), BEDTools (Quinlan and Hall, 2010) and featureCounts (Liao et al., 2014), do not specifically address all three issues. Multireads are not treated specifically by any of these programs and are generally discarded, although this problem has been addressed for ChIP-Seq data analysis (Chung et al., 2011). Ambiguous reads are counted repeatedly for each gene by BEDTools and featureCounts (optionally), whereas HTSeq discards them. HTSeq and featureCounts both consider exon-junction reads, whereas BEDTools does not. ERANGE addresses all three problems, but uses RPKM (reads per kilobase per million) instead of counts throughout the whole algorithm. Moreover, a conversion to counts during the algorithm is not possible (Mortazavi et al., 2008).

Here we describe Rcount, a fast and simple GUI tool for flexible RNA-Seq read counting. It builds on the algorithm described in Schmid et al. (2012), and further allows for editing the genome annotation and assigning priorities to certain feature types (see Figure 1A for details on genomic feature types).

2 DESCRIPTION
Rcount takes read alignments files (BAM, Binary Alignment/Map) and a reference genome annotation (GFF/GTF/BED, General Feature Format/Gene Transfer Format/Browser Extensible Data) as input, and counts the number of reads per gene, taking into account multireads, ambiguous reads and exon-junction reads (Fig. 1). It has three modules: Rcount-multireads, Rcount-format and Rcount-distribute.

Rcount-multireads assigns weights to each alignment of a multiread (Fig. 1B). The weight $H_i$ of an individual alignment $i$ is calculated using a score $S_i$ divided by the sum of scores from all alignments of the multiread ($H_i = S_i / \sum_{i=1}^{n} S_i$). $S_i$ is currently implemented as the sum of coverage (number of reads per base) originating from uniquely aligned reads at the position of the alignment $i$ and the surrounding region (the size can be set by the user). If an alignment spans an exon junction, $S_i$ equals to the number of uniquely aligned reads spanning the same exon junction. Thus, if a multiread has both types of alignments, the ungapped ones are generally preferred. The weight is automatically added as XW:f:H i tag to the alignments in the BAM file.

Rcount-format reads the reference genome annotation in GFF/GTF/BED format, displays the structure of the genome annotation and saves it in an XML format required by Rcount-distribute.

*To whom correspondence should be addressed.
During this process, the user can extend the genes (add flanking regions) or remove features from the annotation. Depending on the library preparation protocol, some of the features in the genome annotation are less likely to be sequenced (e.g. rRNA-coding genes with poly(A)-selective library preparation protocols). The user can choose to remove these features or to assign a lower priority to them. If a read aligns to a location where two genes with different priorities overlap, it is automatically assigned to the one with higher priority.

\textit{Rcount-distribute} sums up the weights of the alignments (hits) per gene in two steps. In the first step, all hits are mapped to all genes (i.e. their transcripts). Transcripts of truly expressed genes should generally have at least some hits in the vicinity of their 3’ end (e.g. due to poly(A)-tail priming during library preparation) and/or at least a minimal total number of hits (user-specified). Transcripts not matching these criteria are discarded during the first round (Fig. 1C). During the second step, the hits are divided into unambiguous and ambiguous. The unambiguous hits are assigned first and subsequently used to proportionally distribute the ambiguous hits (Fig. 1D). The transcripts are re-filtered using the same criteria as before. The final expression value \( c_g \) of a gene is then calculated as the sum of hits assigned to any of its transcripts (Fig. 1D).

The final output is one count table per sample. In addition to the final expression values, the output table also contains the number of unambiguous and ambiguous (before and after distributing them) hits per gene (either on the whole gene length, or only within a certain number of bases from the 3’ end of the transcript, which can be specified by the user). To extract a certain column or to merge multiple samples for downstream analyses, an R script is provided on github.com/MWSchmid/Rcount.

**ACKNOWLEDGEMENT**

We thank Dr. Diana E. Coman Schmid (EAWAG) for helpful discussions and software testing.

**Funding** This work was supported by the University of Zurich, and grants from the Swiss National Science Foundation to U.G.

**Conflict of interest** none declared.

**REFERENCES**


