The Journal of Open Source Software

strandCheckR: An R package for quantifying and removing double strand sequences for strand-specific RNA-seq

Thu-Hien To[1] **and Stephen M Pederson**[1]

1 Bioinformatics Hub - University of Adelaide

DOI: 10.21105/joss.01145

Software

- Review
- Repository
- Archive

Submitted: 14 December 2018
Published: 17 February 2019

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

Summary

RNA-seq data reveals the expression level of every gene/transcript extracted from a specific tissue of an individual. Numerous research questions can involve a study of the relationship between gene expression and phenotypes, or treatment groups. However, assessing data quality is the first step before performing any analysis. `fastqc` (Andrews, 2014) has become an industry-standard quality control tool for much high throughput sequencing data. This package aims to provide an additional quality control step for strand-specific RNA-seq data, to quickly detect issues with library preparation, and also to rescue it in some less extreme cases.

Within the genome, each gene is encoded on either the positive or negative strand, which is used as the template for transcribed, single-stranded RNA. In high-throughput sequencing technology, cellular RNA molecules are extracted, fragmented then several library preparation steps are performed before being sequenced in a massively parallel reaction, which generates millions of reads. A sequencing machine is agnostic to the original strand of the fragments, however, a strand-specific library preparation protocol (Zhong et al., 2011) is able to preserve information about the strand of the original RNA template molecule. Hence, in the bam file, which is obtained from mapping reads back to the reference genome, reads deriving from the same RNA molecule should be mapped to the same strand. In contrast, if contaminating genomic DNA is present, the double stranded nature of these molecules will lead to reads aligning to both strands. By studying the strand of aligned reads within a suitable window, we can identify the presence of reads which are likely to derive from double stranded DNA fragments.

In the R package `strandCheckR`, we use a sliding window to scan any bam files under review. This will return a data frame giving the coordinates of each window and the number of reads aligning to the $+/-$ strands within it. Using this data frame, users can plot a histogram of strand proportions which can enable identification of the levels of DNA contamination. In the example shown in Fig.1, the file `s1.sorted.bam` has minimal contaminating DNA, whilst the file `s2.sorted.bam` shows a pattern consistent with contamination.

We also provide a scatter plot with threshold guidelines (Fig.2) which can help to select an appropriate threshold τ $(0.5 < \tau < 1)$ to filter reads which are likely to derive from contaminating DNA.

For the filtering steps, we use logistic regression to estimate the read strand proportion π_W and its standard deviation σ_W in each window W. We then use these values to decides if W contains any reads coming from RNA or not. The null hypothesis for a window

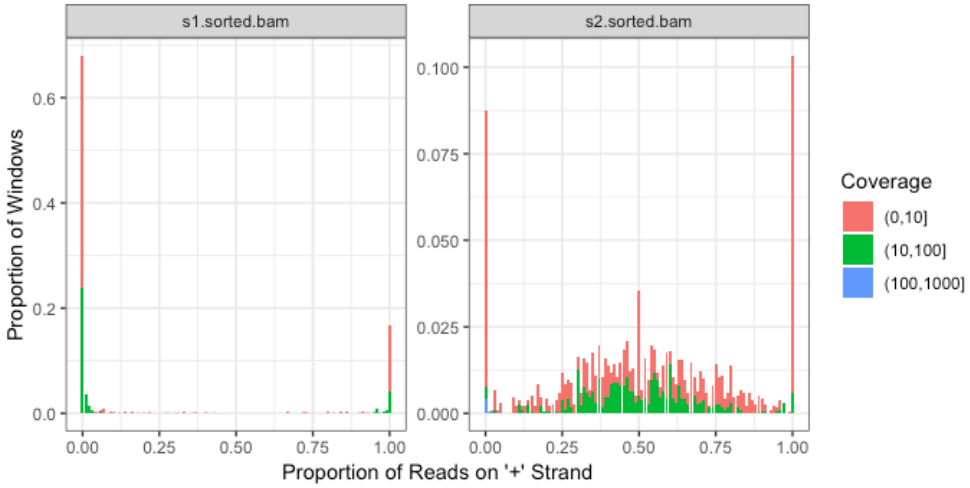

Figure 1: Fig.1 - The file shown on the left appears to have minimal contaminating DNA, whilst the file on the right shows a large number of windows with alignments coming from both strands, indicating potential DNA contamination.

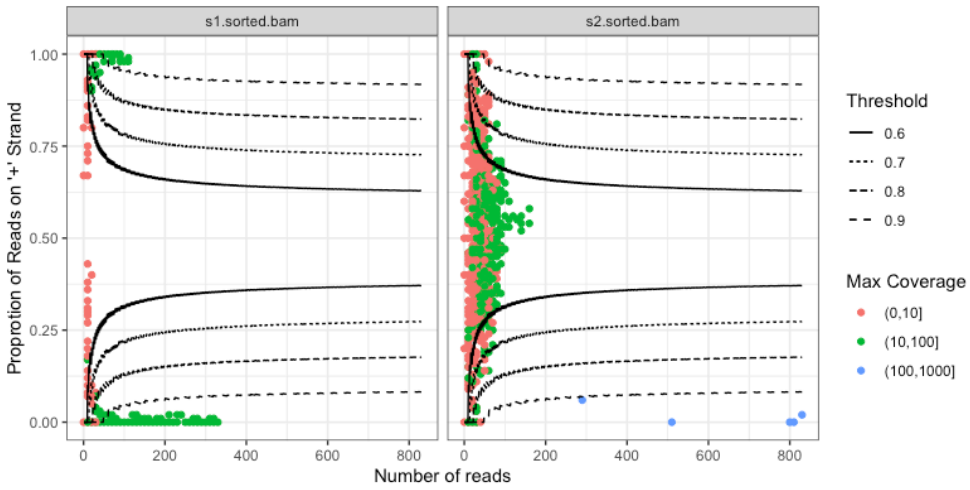

Figure 2: Fig.2 - Plot showing read depth against the proportion of reads aligning to the + strand. Possible values for τ are shown as a series of lines

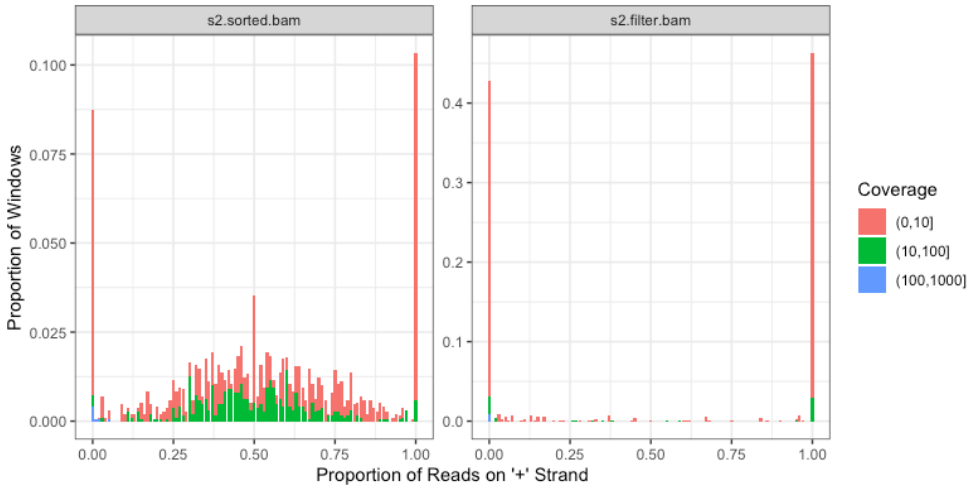

Figure 3: Fig.3 - Histogram showing the contaminated file from Fig 1 (s2.sorted.bam), both before and after filtering reads.

containing positively-stranded reads is $H_0 : \pi < \tau$, and for negative-stranded windows is $H_0 : \pi \geq 1 - \tau$. By default, only windows which reject $H0$ at the level $\alpha = 0.05$ are considered to contain RNA.

For each window W, let p_W and n_W be the number of positive and negative reads in W. We then assign for W an initial probability of keeping positive and negative reads, namely k_W^+ and k_W^- using the following procedure. If W is determined to not contain any read coming from RNA, then $k_W^+ = k_W^- = 0$. Otherwise, W is assumed to contain some RNA reads and two scenarios are investigated. If W contains more positive reads than negative ones ($p_W >> n_W$), then W is assumed to contain RNA coming from positive strand, and we remove reads aligned to the negative strand, (i.e. $k_W^- = 0$) and the same number of reads aligned to the positive strand. As these are assumed to each be paired with a read aligned to the negative strand ones (i.e. $k_W^+ = (p_W - n_W)/p_W$), each positive read is kept with probability k_W^+. The mirror of this process is applied if a windows is determined to contain reads primarily from a negatively stranded gene. However, each read can be belong to several different sliding windows due to overlapping windows can overlap, and due to the nature of spliced alignments. Hence, the final probability of each read is the maximum of all windows which contain that read. Fig.3 shows the differences in strand proportion before and after filtering DNA of an example file.

`strandCheckR` is an R package which make uses of several functions from core Bioconductor packages such as `GenomicAlignments`, `GenomicRanges` (Lawrence et al., 2013), and `Rsamtools` (Morgan, Pagès, Obenchain, & Hayden, 2018). The window read count function is designed flexibly so that user can filter low mapping quality reads, set the minimum proportion required for a read to overlap and be included in a window, define window length & step size etc. It has also been implemented in an efficient way to manage large bam files. For a typical human RNA-seq bam file, it takes about 3 minutes to scan and get strand information using a standard laptop 2,3 GHz i5 16 GB. The package can be installed via Bioconductor repository https://bioconductor.org/packages/strandCheckR and is also available on github http://github.com/UofABioinformaticsHub/strandCheckR.

References

Andrews, S. (2014). *FastQC a quality control tool for high throughput sequence data*. Retrieved from https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M., et al. (2013). Software for computing and annotating genomic ranges. *PLoS Computational Biology*, *9*(8). doi:10.1371/journal.pcbi.1003118

Morgan, M., Pagès, H., Obenchain, V., & Hayden, N. (2018). *Rsamtools: Binary alignment (bam), fasta, variant call (bcf), and tabix file import*. Retrieved from http://bioconductor.org/packages/release/bioc/html/Rsamtools.html

Zhong, S., Joung, J.-G., Zheng, Y., Chen, Y.-r., Liu, B., Shao, Y., Xiang, J. Z., et al. (2011). High-throughput illumina strand-specific rna sequencing library preparation. *Cold Spring Harbor Protocols*, *2011*(8), pdb.prot5652. doi:10.1101/pdb.prot5652