SPARTA: Simple Program for Automated reference-based bacterial RNA-seq Transcriptome Analysis

Benjamin K. Johnson[1], Matthew B. Scholz[2], Tracy K. Teal[3], and Robert B. Abramovitch[1]*

[1]Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, Michigan, 48824, USA, [2]Institute for Cyber-Enabled Research, Michigan State University, East Lansing, Michigan, 48824, USA, [3]Data Carpentry.

*Corresponding author

ABSTRACT

Summary: SPARTA is a reference-based bacterial RNA-seq analysis workflow application for single-end Illumina reads. SPARTA is turnkey software that simplifies the process of analyzing RNA-seq data sets, making bacterial RNA-seq analysis a routine process that can be undertaken on a personal computer or in the classroom. The easy-to-install, complete workflow processes whole transcriptome shotgun sequencing data files by trimming reads and removing adapters, mapping reads to a reference, counting gene features, calculating differential gene expression, and, importantly, checking for potential batch effects within the data set. SPARTA outputs quality analysis reports, gene feature counts and differential gene expression tables and scatterplots. The workflow is implemented in Python for file management and sequential execution of each analysis step and is available for Mac OS X, Microsoft Windows, and Linux. To promote the use of SPARTA as a teaching platform, a web-based tutorial is available explaining how RNA-seq data are processed and analyzed by the software.

Availability and Implementation: Tutorial and workflow can be found at sparta.readthedocs.org. Teaching materials are located at sparta-teaching.readthedocs.org. Source code can be downloaded at www.github.com/abramovitchMSU/, implemented in Python and supported on Mac OS X, Linux, and MS Windows.

Contact: Robert B. Abramovitch (abramov5@msu.edu)

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

1 INTRODUCTION

One of the most common applications of RNA sequencing (RNA-seq) is to identify differentially expressed genes under differing experimental conditions. Before biological insights can be gained, one must process and analyze the large datasets generated from each sequencing experiment. Each sample contains millions of reads that must be trimmed and assessed for read quality, mapped back to a reference genome (or assembled *de novo* in the absence of a reference), counted for transcript abundance, and tested for differential gene expression. Many computational analysis tools have been developed specifically to work with RNA-seq data; however, a single tool is often not suitable and requires several different applications assembled into a workflow. This task can be complicated as both the tool choice and input and output file formats for a given tool need to be considered and potentially modified to meet the requirements for the subsequent analysis step. Several RNA-seq analysis workflows exist, however, most are designed for eukaryotic organisms (D'Antonio, et al., 2015; Golosova, et al., 2014; Goncalves, et al., 2011; Habegger, et al., 2011; Kalari, et al., 2014; McClure, et al., 2013; Michalovova, et al., 2015; Qi, et al., 2010; Tjaden, 2015; Wang, et al., 2011). The goal of this work is to assemble several open-source computational tools to deliver a complete, accessible, and easy-to-use reference-based bacterial RNA-seq analysis workflow that is amenable to both the research laboratory and undergraduate classroom.

2 FEATURES AND FUNCTIONS

The SPARTA workflow (Figure 1) is implemented utilizing Python for file input/output management and tool execution, combining several open-source computational tools. The SPARTA workflow analyzes data by: conducting read trimming and adapter removal with Trimmomatic (Bolger, et al., 2014); performing quality analysis of the data sets with FastQC (Anders, 2010); mapping the reads to the reference with Bowtie (Langmead, et al., 2009); counting transcript or gene feature abundance with HTSeq (Anders, et al., 2015); and, analyzing differential gene expression with edgeR (McCarthy, et al., 2012; Robinson, et al., 2010; Robinson and Oshlack, 2010). Within the differential gene expression analysis step, batch effects can be detected and the user is warned that potentially unintended variables need to be considered. If left unaccounted for, batch effects can significantly skew the results of the data analysis, leading to inappropriate experimental conclusions (Leek, et al., 2010). Following analysis, SPARTA outputs quality analysis reports, gene feature counts and differential gene expression tables and scatterplots.

SPARTA requires Python 2, NumPy (a Python library for numerical analyses), Java and R. Once Python is installed, the user initializes SPARTA, which then checks for the necessary dependencies at runtime. If any of these dependencies are not met, SPARTA informs the user of the missing components. To reduce complex software installation, SPARTA is distributed with the required software and an online tutorial (sparta.readthedocs.org) guides the user through installation and data analysis procedures for each operating system platform. The workflow maintains analytic flexibility for specific use cases by allowing the user to tailor the options utilized for each

analysis step, but can proceed without requiring option specification. Further, SPARTA will write the necessary R commands at runtime and will generate the appropriate contrasts to test all possible comparisons between user defined experimental conditions. Using a previously published data set (Baker, et al., 2014), SPARTA was capable of analyzing 4 experimental conditions containing 8 samples with approximately 30 million reads per sample in 4 hours on an off-the-shelf iMac computer (8 GB RAM, Intel i5 2.7GHz quad-core processor). SPARTA can also be implemented in high performance computing environments utilizing the non-interactive mode functionality (Supplemental material).

As NGS technologies and applications continue to permeate life science research, undergraduate education must include the use of contemporary sequencing techniques to address biological questions. However, despite the rapid increase in data intensive experimental biology, undergraduates receiving a life sciences degree are often not exposed to the tools and basic computational skills required to study NGS data sets. To address this shortcoming, we have developed an online tutorial to guide students through the RNA-seq analysis process (sparta-teaching.readthedocs.org). The SPARTA teaching tool was integrated into a senior level genomics course and successfully engaged students in the theory and application of RNA-seq data analysis.

In summary, RNA-seq transcriptional profiling is becoming increasingly routine, and there is a demand for applications such as SPARTA that enable stand-alone workflows. SPARTA represents an easy-to-use, platform independent analysis workflow for reference-based bacterial RNA-seq amenable to the research laboratory and classroom.

ACKNOWLEDGEMENTS

This project was supported by grants to RBA from the NIH (R21AI105867) and the Bill & Melinda Gates Foundation (OPP1119065).

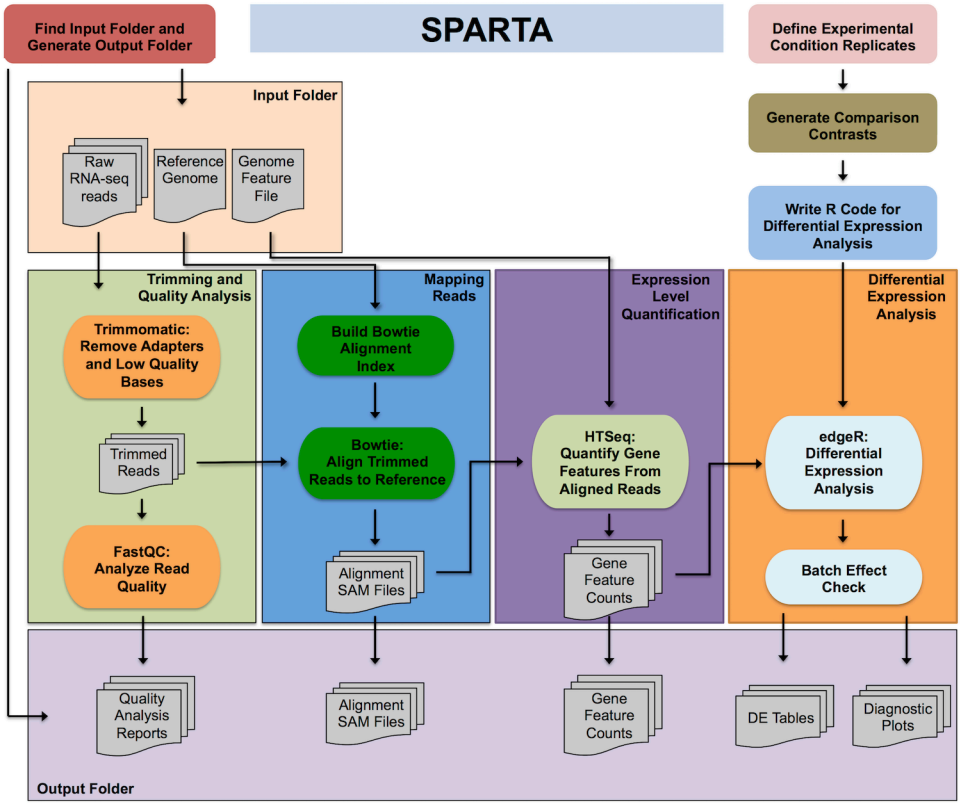

Figure 1. SPARTA workflow diagram

REFERENCES

Anders, S. (2010) FastQC - http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
Anders, S., Pyl, P.T. and Huber, W. (2015) HTSeq--a Python framework to work with high-throughput sequencing data, *Bioinformatics*, 31, 166-169.
Baker, J.J., Johnson, B.K. and Abramovitch, R.B. (2014) Slow growth of Mycobacterium tuberculosis at acidic pH is regulated by phoPR and host-associated carbon sources, *Mol Microbiol*, 94, 56-59.
Bolger, A.M., Lohse, M. and Usadel, B. (2014) Trimmomatic: a flexible trimmer for Illumina sequence data, *Bioinformatics*, 30, 2114-2120.
D'Antonio, M.*, et al.* (2015) RAP: RNA-Seq Analysis Pipeline, a new cloud-based NGS web application, *BMC genomics*, 16, S3.
Golosova, O.*, et al.* (2014) Unipro UGENE NGS pipelines and components for variant calling, RNA-seq and ChIP-seq data analyses, *PeerJ*, 2, e644.
Goncalves, A.*, et al.* (2011) A pipeline for RNA-seq data processing and quality assessment, *Bioinformatics*, 27, 867-869.
Habegger, L.*, et al.* (2011) RSEQtools: a modular framework to analyze RNA-Seq data using compact, anonymized data summaries, *Bioinformatics*, 27, 281-283.
Kalari, K.R.*, et al.* (2014) MAP-RSeq: Mayo Analysis Pipeline for RNA sequencing, *BMC bioinformatics*, 15, 224.
Langmead, B.*, et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome Biol*, 10, R25.
Leek, J.T.*, et al.* (2010) Tackling the widespread and critical impact of batch effects in high-throughput data, *Nature reviews. Genetics*, 11, 733-739.
McCarthy, D.J., Chen, Y. and Smyth, G.K. (2012) Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation, *Nucleic Acids Res*, 40, 4288-4297.
McClure, R.*, et al.* (2013) Computational analysis of bacterial RNA-Seq data, *Nucleic Acids Res*, 41, e140.
Michalovova, M.*, et al.* (2015) Fully automated pipeline for detection of sex linked genes using RNA-Seq data, *BMC bioinformatics*, 16, 78.
Qi, J.*, et al.* (2010) inGAP: an integrated next-generation genome analysis pipeline, *Bioinformatics*, 26, 127-129.
Robinson, M.D., McCarthy, D.J. and Smyth, G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data, *Bioinformatics*, 26, 139-140.
Robinson, M.D. and Oshlack, A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biol*, 11, R25.
Tjaden, B. (2015) De novo assembly of bacterial transcriptomes from RNA-seq data, *Genome Biol*, 16, 1.
Wang, Y.*, et al.* (2011) RseqFlow: workflows for RNA-Seq data analysis, *Bioinformatics*, 27, 2598-2600.