Sequence analysis

aRNApipe: A balanced, efficient and distributed pipeline for processing RNA-seq data in high performance computing environments

Arnald Alonso[1,2], Brittany N. Lasseigne[1], Kelly Williams[1], Josh Nielsen[1], Ryne C. Ramaker[1,3], Andrew A. Hardigan[1,3], Bobbi Johnston[1], Brian S. Roberts[1], Sara J. Cooper[1], Sara Marsal[2] and Richard M. Myers[1,*]

[1]HudsonAlpha Institute for Biotechnology, Huntsville, AL 35806, USA.

[2]Rheumatology Research Group, Vall d'Hebron Hospital Research Institute, Barcelona, Spain

[3]Department of Genetics, The University of Alabama at Birmingham, Birmingham, AL 35294, USA.

*To whom correspondence should be addressed.

Abstract

Summary: The wide range of RNA-seq applications and their high computational needs require the development of pipelines orchestrating the entire workflow and optimizing usage of available computational resources. We present aRNApipe, a project-oriented pipeline for processing of RNA-seq data in high performance cluster environments. aRNApipe is highly modular and can be easily migrated to any high performance computing (HPC) environment. The current applications included in aRNApipe combine the essential RNA-seq primary analyses, including quality control metrics, transcript alignment, count generation, transcript fusion identification, and sequence variant calling. aRNApipe is project-oriented and dynamic so users can easily update analyses to include or exclude samples or enable additional processing modules. Workflow parameters are easily set using a single configuration file that provides centralized tracking of all analytical processes. Finally, aRNApipe incorporates interactive web reports for sample tracking and a tool for managing the genome assemblies available to perform an analysis.

Availability and documentation: https://github.com/HudsonAlpha/aRNAPipe

Contact: rmyers@hudsonalpha.org

Supplementary information: Supplementary data are available.

1 Introduction

Quantification of RNA transcripts by next-generation sequencing technologies continues to increase in both throughput and capabilities year after year as sequencing becomes more affordable and accessible (McGettigan, 2013). Unlike gene expression microarrays, RNA-seq not only quantifies gene expression levels, but also allows measurement of alternative splicing, transcript fusions, and RNA sequence variants (Finotello and Di Camillo, 2015; Koboldt, et al., 2012; Maher, et al., 2009). This broad spectrum of applications has fostered development of a rich set of bioinformatics methods focused on each stage of the processing workflow (Conesa, et al., 2016). Current applications for primary analysis of RNA-seq data usually apply only a single processing step, involve complex dependencies between data processing stages, and depend on the sequencing protocol performed. Consequently, there is an increasing need for tools orchestrating the entire analysis workflow to ensure repeatability of RNA-seq data processing.

In addition to the need for data processing integration, the computational requirements of some RNA-seq analysis steps are a bottleneck for studies involving large numbers of samples (Scholz, et al., 2012). For these studies, the use of high performance computing (HPC) clusters is unavoidable. Because HPC clusters are a valuable and often limited resource, tools integrating the different stages of RNA-seq processing must be carefully designed and optimized.

Considering these challenges, we have developed a balanced, efficient and distributed pipeline for analysis of RNA-seq data: aRNApipe (**a**utomated **RNA**-seq **pipe**line). This pipeline has been optimized to efficiently exploit the resources of HPC clusters by processing sample batches together through each analysis stage. This allows users to balance the computational resources of each processing step and to scale easily from tens to thousands of RNA-seq libraries. aRNApipe is a highly modular tool that orchestrates the entire processing workflow providing interactive reports at each stage, as well as efficient and traceable parameterization of each processing step.

2 Methods

aRNApipe has been designed to overcome the challenges of integration, synchronization and reporting of RNA-seq data analysis by using a project-oriented and balanced design optimized for HPC clusters (Fig. 1). The balanced design allows independent assignment of the HPC resources used at each processing step, thereby using less resources for the less intense steps.

The core application of aRNApipe (Supplementary Section S1) includes six operating modes: 1) executing a new analysis, 2) updating a previous analysis to include new samples or enable new modules, 3) showing progress of an analysis, 4) building a skeleton for a project, 5) showing available genome builds, and 6) stopping an ongoing analysis.

Input data: aRNApipe requires only two input files: 1) analysis configuration, and 2) samples to include in the analysis. In the configuration file, the user can set the executing parameters, including enabling processing modules, assigning computational resources to each module (memory and number of CPUs), selecting the reference genome build, and designating the arguments of each module. The sample file contains both the sample identifiers and the paths to corresponding raw data files.

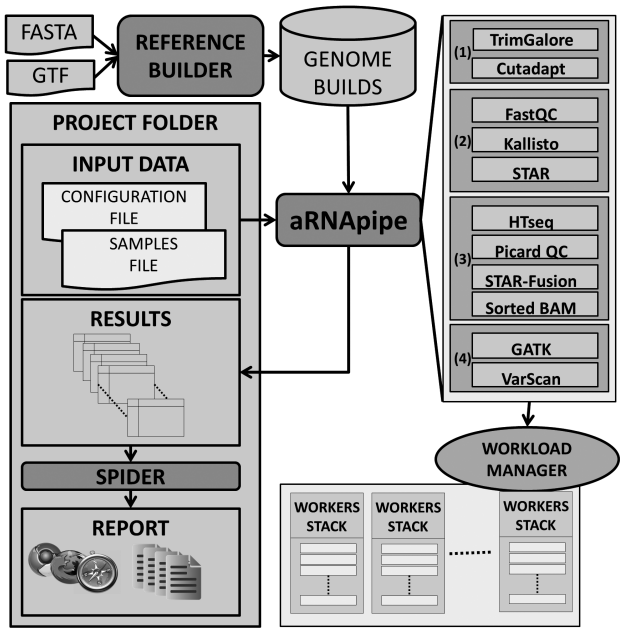
Current applications: aRNApipe currently includes state-of-the-art applications covering the main variations of RNA-seq data generation (Supplementary Section S2). Throughout the workflow, a main daemon process manages pipeline execution (i.e. inter-dependencies between applications) and monitors analysis of each sample at each stage (Fig. 1). First, low-quality reads/bases and adapter sequences can be filtered. After the first stage, if enabled, a second stack of applications is run in parallel, including assessment of raw data quality, transcriptome pseudo-alignment and transcript quantification and alignment to a reference genome. The main process, which manages cross-application dependencies, waits until STAR analysis is completed to launch a third stack of analyses including identification of gene fusions, quantification of genes and exons, conversion of SAM files to BAM sorted files, and assessment of alignment quality. Finally, once BAM sorted files are ready, a fourth stack of applications including two variant calling modules are run.

Report generation: The Spider is an aRNApipe add-on module that generates interactive web reports summarizing an aRNApipe analysis. It can be run on project currently being processed or a project that has been previously processed. These reports provide a comprehensive overview of the results of each module, including sample quality control metrics and basic statistics like principal component analysis and the distribution of gene/transcript count data (Supplementary Figures 1 and 2). The user can also obtain information about the computational resources used by each module and access to all logs generated during analysis (Supplementary Figure 3). Another important Spider feature is the generation of matrix-like count data files of raw counts, RPKMs, and their corresponding annotation files (i.e. gene identifier and length) for all project samples to facilitate downstream differential expression analysis (Love, et al., 2014; Nikolayeva and Robinson, 2014). Supplementary Section S3 provides a list of all the outputs generated by the report generation tool.

Reference builder: The programs used for RNA-seq data processing often use different formats and standards for their input reference and annotation files. To address the problems associated with the lack of a consensus file format and to provide a centralized repository of available genome builds, aRNApipe provides a reference builder that generates all required files for using a genome build based on a set of initial files that can be easily obtained from sources such as the NCBI and the Ensembl repositories (Supplementary Section S4 and Supplementary Figure 4).

Implementation: aRNApipe has been developed using Python 2.7 (Supplementary Figure 5). When running on an HPC cluster, aRNApipe relies on the workload management application to submit the jobs for each processing stage, taking into account cross-stage dependencies and using custom resource requirements for each stage. aRNApipe has been implemented with the workload management system IBM Platform LSF, but its design allows quick migration to any other workload manager by editing one Python library containing five simple workload manager dependent functions (Supplementary Section S5). Additionally, a single-machine version is also provided to run the pipeline on a computer with at least 40GB of RAM memory. A configuration library provides a quick and simple way to supply paths to all applications used by aRNApipe.

Fig. 1. aRNApipe workflow for primary analysis of RNA-seq data



3 Results

We have extensively tested aRNApipe and used it to analyze hundreds of RNA-seq libraries with multiple configurations, including different species, different genome builds and different RNA-seq protocols. The reports generated for four example datasets can be accessed online (http://arnapipe.bitbucket.org): (1) Strand-specific paired-end RNA-seq data from 9 human samples of different tissues (GSE69241), (2) unstranded single-end data from two paired normal and colorectal tumor tissues (GSE29580), (3) unstranded paired-end data from 11 melanoma samples (GSE20156) and 1 prostate cancer cell line (NCIH660), and (4) in-house unstranded paired-end data from 20 zebrafish libraries.

4 Conclusions

aRNApipe provides an integrated and efficient workflow for analyzing single-end and stranded or unstranded paired-end RNA-seq data. Unlike previous pipelines , aRNApipe is focused on HPC environments and the independent designation of computational resources at each stage allow optimization of HPC resources. This application is highly flexible because its project configuration and management options. The Spider provides functional reports for the user at all analytical stages and the reference builder is a valuable genome build manager. Finally, aRNApipe's modularity allows it to be adapted to changes in the current applications and the addition of new functionalities. Implementation of this pipeline allows users to quickly and efficiently complete primary RNA-seq analysis.

Acknowledgements

We thank all the members of the Myers Lab at the HudsonAlpha Institute of Biotechnology for their constructive suggestions, feedback and testing during the implementation of this pipeline. We also thank the HudsonAlpha IT team for their continuous support during the development of the pipeline and its optimization on the HPC cluster.

Conflict of Interest: none declared.

References

Conesa, A., *et al.* (2016) A survey of best practices for RNA-seq data analysis, *Genome Biology*, **17**, 1-19.

Finotello, F. and Di Camillo, B. (2015) Measuring differential gene expression with RNA-seq: challenges and strategies for data analysis, *Briefings in Functional Genomics*, **14**, 130-142.

Koboldt, D.C., *et al.* (2012) VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing, *Genome Research*, **22**, 568-576.

Love, M.I., Huber, W. and Anders, S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2, *Genome Biology*, **15**, 1-21.

Maher, C.A., *et al.* (2009) Transcriptome sequencing to detect gene fusions in cancer, *Nature*, **458**, 97-101.

McGettigan, P.A. (2013) Transcriptomics in the RNA-seq era, *Current Opinion in Chemical Biology*, **17**, 4-11.

Nikolayeva, O. and Robinson, M.D. (2014) edgeR for differential RNA-seq and ChIP-seq analysis: an application to stem cell biology, *Stem Cell Transcriptional Networks: Methods and Protocols*, 45-79.

Scholz, M.B., Lo, C.-C. and Chain, P.S.G. (2012) Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis, *Current Opinion in Biotechnology*, **23**, 9-15.