Computational Considerations in Transcriptome Assemblies and Their Evaluation, using High Quality Human RNA-Seq data

Noushin Ghaffari
AgriLife Genomics and Bioinformatics,Texas A&M AgriLife Research
Center for Bioinformatics and Genomic Systems Engineering (CBGSE)
101 Gateway Blv.
College Station, TX 77845, US
nghaffari@tamu.edu

Jordi Abante
AgriLife Genomics and Bioinformatics,Texas A&M AgriLife Research
Center for Bioinformatics and Genomic Systems Engineering (CBGSE)
101 Gateway Blv.
College Station, TX 77845, US
jordi.abante@tamu.edu

Raminder Singh
Indiana University Research Technologies
2709 East 10th Street
Bloomington, IN 47408, US
ramifnu@iu.edu

Philip D. Blood
Pittsburgh Supercomputing Center
300 S Craig St.
Pittsburgh, PA 15213, US
blood@psc.edu

Charles D. Johnson
AgriLife Genomics and Bioinformatics,Texas A&M AgriLife Research
Center for Bioinformatics and Genomic Systems Engineering (CBGSE)
101 Gateway Blv.
College Station, TX 77845, US
charlie@ag.tamu.edu

ABSTRACT

It is crucial to understand the performance of transcriptome assemblies to improve current practices. Investigating the factors that affect a transcriptome assembly is very important and is the primary goal of our project. To that end, we designed a multi-step pipeline consisting of variety of pre-processing and quality control steps. XSEDE allocations enabled us to achieve the computational demands of the project. The high memory Blacklight, Greenfield, and Bridges systems at the Pittsburgh Supercomputing Center (PSC) were essential to accomplish multiple steps of this project. This paper presents the computational aspects of our comprehensive transcriptome assembly and validation study.

CCS Concepts

•**Applied computing** → **Computational transcriptomics;** *Sequencing and genotyping technologies; Bioinformatics;* Computational genomics;

Keywords

RNA-Seq; Next-generation sequencing; Transcriptome assembly; Quality control, High performance computing, Data management challenges.

XSEDE16, July 17 - 21, 2016, ,

ISBN 978-1-4503-4755-6/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2949550.2949572

1. INTRODUCTION

In the last decade, genomic sequencing technology has advanced significantly. The data analysis tools for resulting sequencing data has grown simultaneously. However, it is important to choose the correct tool for the application and validate the results. This paper presents computational aspects of our proposed workflow for transcriptome assembly and investigating the results thoroughly. We propose using the Sequencing Quality Control (SEQC) Consortium RNA-Seq data [2] for assembling human transcriptome. The comparison of our resulting *de novo* assembled transcriptomes with the well-annotated human transcriptome can provide insights to the pros and cons of the assembly procedure. The SEQC data was generated at multiple sequencing centers/sites and included well-studied samples, coordinated by the US Food and Drug Administration (FDA). The SEQC sample A (UHRR) consists of ten pooled cancer cell lines, and sample B (HBRR) is from multiple brain regions of 23 donors. We selected samples A and B (sample-type effect), across all six sequencing sites (site effect), and assemblies will be done for different number of pooled replicates (sequencing depth effect).

Our study is the most comprehensive transcriptome assembly and quality control study with emphasis on evaluations to the best of our knowledge. The project aims

to generate assemblies, with fixed parameters, to examine the effects of built-in settings of SEQC data on the results. So far XSEDE [14] allocations and Extended Collaborative Support Service (ECSS) had been an invaluable resource in completing first steps of the project. Upon completion of the project we will report our results in a reputable journal to assist scientific community with recommendations on best practices for transcriptome assembly. There are three important aspects of the study: I- Site effect examinations, II- Coverage/Sequencing depth effect examinations, III- Library effect examinations.

2. METHODS

The assembly and evaluation process includes multiple quality assessment steps. Each assembly goes through a comprehensive pipeline of quality measurement tools to ensure the satisfactory quality and to rank the outputs. The ECSS enables development of a workflow for performing assemblies, and quality control of case studies. This section provides the computational detail of our workflow and usage of the software as well as the computational challenges that we are facing.

2.1 Software Pipeline

All the tools that are part of our pipeline can be found in Table 1. This table presents the tools in the order that were called and can serve as a study outline as well. Before any further processing of the data, the RNA-seq reads have to go through the pre-processing phase. The most frequent format for sequenced fragments, called reads, is referred to as *FASTQ* format. Each read consists of four lines in such a file, beginning with the first line as an identifier of the read. The second line contains the sequence itself, i.e. the string of nucleotides that the sequencing machine reads. The third line is the character plus (+), and the fourth line contains a quality score for each base called by the sequencing machine. We use this information along several tools to improve the quality of the data used for the downstream processing and analysis.

To pre-process, we use four different tools sequentially, mainly: Cutadapt [11], Flexbar [3], Bowtie [7] and SEECER [8]. We use Cutadapt to eliminate the adapters added to the sequences for sequencing purposes. In order to remove the poly A/T tails, added to the mRNA right after transcription, we use Flexbar. With the purpose of removing reads coming from ribosomal RNA (rRNA) and the mitochondrial chromosome, we align the reads to their respective references and we discard the reads that actually align. The reads that pass all the previous filters are then fed to SEECER, a sequencing error correction algorithm for RNA-seq data sets. The pre-processing steps used less than 5% of the total time of our pipeline. However, all the downstream tools and analysis can benefit from their effects on the input data, in terms of quality of the transcriptome assemblies and the reliability of the output. More details on the benefits of different parts of the workflow and the scientific findings of the study will be provided in a manuscript soon, upon completion of all the steps.

Once the reads are ready for downstream analysis, we used Trinity2 [5] to assemble the transcriptomes. There are many transcriptome assembly tools available, and although most of those generate reliable results, we chose Trinity assembler. In another study, we compared the performance of assembly

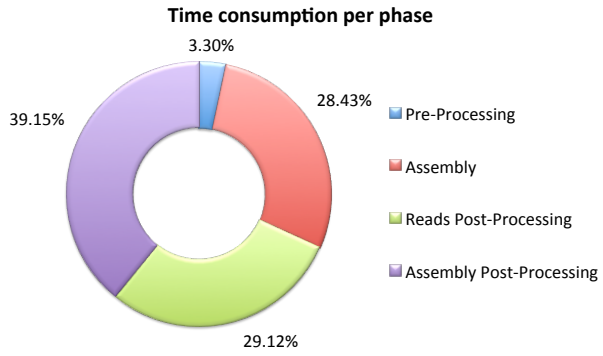

Figure 1: Time consumption per phase up to April 2016.

tools and reported that Trinity outperforms other software, especially in statistical evaluations such as contiguity of the transcripts [4]. Furthermore, other studies concluded that Trinity performs the best [1]. The computational requirements of the assembling process are highly demanding, especially in terms of Random Access Memory (RAM). We found it best to run Trinity on 30 CPUs on Greenfield (machine specific) to provide enough memory and disk space for each assembly run. We derived the number of CPUs based on test runs and considerations of the input data volume. For efficiency and reliability, we configured job scripts to move data to RAMDISK on the compute nodes and use RAMDISK as runtime persistence location. Job scripts moved the data to project location after completion of a run. The files resulting of the assembly process are in FASTA format. This format consists of two lines for each assembled sequence, called a contig. The first line is the contig identifier while the second line is the sequence of nucleotides that form the contig.

One of our main goals is to understand factors that have a significant impact in the transcriptome assembly outcome. We assess the quality of the transcriptomes in multiple ways. We use BUSCO [13] to look for the number of single-copy orthologs coming from a vertebrate gene database present in the transcriptome. That allows us to assess the completeness of the transcriptome from a gene content perspective. We use DETONATE [9] to study the k-mer proportions present in the transcriptome compared to that in the reference. The computational requirements of DETONATE are highly demanding, being the step that takes the most time among completed steps (larger assemblies will ran later and might affect the most consuming tool report).

Another metric that we use in order to assess the quality of the transcriptomes is the coverage when mapping them back to the reference. We use GMAP [15], a tool originally conceived specifically for that purpose that outperforms other aligners when the query sequences are rather long. On the other hand, we align the reads used to build the transcriptomes to the reference genome using Tophat2 [6]. The files obtained are in SAM format [10] in both cases. The coverage is then compared between the alignment of the transcriptome and the alignment of the respective reads using Samtools [10].

Once the transcriptomes are aligned back to the reference, we use GATK [12] to look for single-base nucleotide polymorphisms (SNPs) in the alignments. We also run GATK on the reads alignments and compare the SNP calls between the

Phase	Tool	Purpose	CPU	GB/CPU	Time(min.)	Dependencies
Pre-processing	Cutadapt	Get rid of adapters in the reads.	16	8	50	Python/2.7.9
	Flexbar	Get rid of poly A/T tails in the reads.	15	8	90	Samtools/1.2, Flexbar/2.5
	Bowtie	Remove reads coming from rRNA and chrM in the reads.	15	8	90	Samtools/1.2, Bowtie/1.1.1
	SEECER	Correct errors in reads to improve transcriptome assembly.	15	24	60	SEECER/0.1.3
Reads post-processing	Tophat2	Map the reads to the reference genome.	15	8	2040	Tophat/2.1.0-all
	GenomeCoverageBed	Compute Tophat's output genome coverage.	4	4	20	Samtools
	Samtools sort	Sort Tophat's output BAM file kariotypically.	5	4	250	Samtools
	CreateSequenceDictionary	Generate dictionary from reference.	1	2	10	Picard
	GATK	Call single-nucleotide polymorphisms (SNPs) in Tophat's output.	4	4	240	Java, Samtools, Picard
Assembly and post-processing	Trinity	Assemble transcriptomes.		66	2500	Trinity/2.0.6-all
	BUSCO	Assess transcriptome completeness with single-copy orthologs.	8	2	80	EMBOSS, HMMER, NCBI-blast, Python
	DETONATE	Evaluate transcriptome assemblies.	15	8	2700	DETONATE, Bowtie2
	GMAP build	Builds a GMAP database for the reference genome.	4	4	20	GMAP
	GMAP	Map the transcriptome contigs to the reference genome.	8	4	60	GMAP
	GenomeCoverageBed	Compute Trinity's output genome coverage.	4	4	20	Samtools
	Samtools sort	Sort BAM file kariotypically.	1	2	300	Samtools
	Samtools index	Generate indexes.	1	2	10	Samtools
	CreateSequenceDictionary	Generate dictionary from reference.	1	2	10	Picard
	Samtools faidx	Index reference.	1	2	2	Samtools
	GATK	Call single-nucleotide polymorphisms (SNPs) in transcriptome.	4	4	240	Java, Samtools, Picard

Table 1: Tools used and their computational requirements for all the different phases of our transcriptome assembly and QC pipeline. All the tools ran on Greenfield system at PSC, except the Cutadapt. The Cutadapt ran on PSC Blacklight.

transcriptome alignment and the respective reads alignment. Before running GATK, the SAM files have to be sorted karyotypically and have to be indexed using Samtools. As far as the reference is concerned, in order to run GATK, it has to be indexed and a dictionary has to be generated using Picard tools. Once the SNPs are called, these are stored in variant call format (VCF). Figure 1 shows the amount of time consumed for each phase of the project up to this point. As the figure presents, the assembly used up approximately third of the total time. The assembly post-processing consumed almost a 40% of the total time. The most resources in the last phase are used by the DETONATE tool.

2.2 Computational challenges

Analyzing many terabytes of data through a complex pipeline presents many challenges. Here we briefly discuss some of those challenges:

Data Movement

Data must move back and forth from long-term storage to Scratch space, and also to RAMDISK for processing. Data management is challenging because of the multiple locations of the data. Compute nodes do not always have direct access to long-term storage, which makes the movement a multistep workflow. Having allocated long-term storage space connected to compute nodes, as is done with the new Bridges system, will help address this challenge.

Data Verification

Each step needs data verification before processing the next steps. For example, while pre-processing we found that sometimes reads were missing and the number of left and right reads (forward and backward sequences pairs) did not match. To address this we have added an input verification step before any further processing.

Checkpointing and Recovery

This workflow involves long running processes with little or no support for recovery. Future work may explore the use of external checkpointing mechanisms for some of these workflow steps, such as the SLURM-integrated Berkeley Lab Checkpoint/Restart (BLCR) functionality.

Certain programs with some checkpointing capability, like Trinity, run best in RAMDISK. However, by design the RAMDISK space is purged on completion or failure of the job. In this case the run output needs to be periodically synced to the shared filesystem and verified to ensure that progress of long-running jobs is not lost.

3. CONCLUSIONS

The current project aims to run quality controls and assemblies using a reliable SEQC data to study the factors that affect the results. To achieve this goal, the XSEDE allocations enabled us to run a multi-step workflow, consisting of numerous tools. The high memory Blacklight, Greenfield, and Bridges systems at PSC were essential to accomplish multiple steps of this project. As described above, each step faced challenges and with the help of ECSS most tasks of the project have been completed. Upon completion of the

project, we intend to publish our scientific findings in a journal.

4. ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. Specifically, it used the Bridges, Greenfield, and Blacklight systems, which are supported by NSF award numbers ACI-1445606, ACI-1261721, and ACI-1041726, respectively, at the Pittsburgh Supercomputing Center (PSC). This material is also based upon work supported by the National Science Foundation under Grant Number DBI-1458689.

5. REFERENCES

[1] A. Celaj, J. Markle, J. Danska, and J. Parkinson. Comparison of assembly algorithms for improving rate of metatranscriptomic functional annotation. *Microbiome*, 2(39), 2014.

[2] SEQC/MAQC-Iii. Consortium et al. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the sequencing quality control consortium. *Nature biotechnology*, 32(9):903–914, 2014.

[3] M. Dodt, J. T. Roehr, R. Ahmed, and C. Dieterich. FLEXBAR - flexible barcode and adapter processing for next-generation sequencing platforms. *Biology*, 1(3):895–905, 2012.

[4] N. Ghaffari, O. A. Arshad, H. Jeong, J. Thiltges, M. F. Criscitiello, B.-J. Yoon, A. Datta, and C. D. Johnson. Examining de novo transcriptome assemblies via a quality assessment pipeline. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1), 2015.

[5] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, et al. Trinity: reconstructing a full-length transcriptome without a genome from RNA-seq data. *Nature biotechnology*, 29(7):644, 2011.

[6] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, S. L. Salzberg, et al. Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*, 14(4):R36, 2013.

[7] B. Langmead, C. Trapnell, M. Pop, S. L. Salzberg, et al. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biol*, 10(3):R25, 2009.

[8] H.-S. Le, M. H. Schulz, B. M. McCauley, V. F. Hinman, and Z. Bar-Joseph. Probabilistic error correction for RNA sequencing. *Nucleic acids research*, page gkt215, 2013.

[9] B. Li, N. Fillmore, Y. Bai, M. Collins, J. A. Thomson, R. Stewart, and C. N. Dewey. Evaluation of de novo transcriptome assemblies from rna-seq data. *Genome Biol*, 15(12):553, 2014.

[10] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[11] M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):pp–10, 2011.

[12] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research*, 20(9):1297–1303, 2010.

[13] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19):3210–3212, 2015.

[14] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, et al. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74, 2014.

[15] T. D. Wu and C. K. Watanabe. Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics*, 21(9):1859–1875, 2005.