Cyberinfrastructure resources enabling creation of the loblolly pine reference transcriptome

Le-Shin Wu
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
lewu@iu.edu

Carrie L. Ganote
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
cganote@iu.edu

Thomas G. Doak
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
tdoak@iu.edu

William Barnett
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
barnettw@iu.edu

Keithanne Mockaitis
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
kmockait@indiana.edu

Craig A. Stewart
Indiana University
Pervasive Technology Institute
2709 E. Tenth Street
Bloomington, IN 47408
stewart@iu.edu

ABSTRACT

Today's genomics technologies generate more sequence data than ever before possible, and at substantially lower costs, serving researchers across biological disciplines in transformative ways. Building transcriptome assemblies from RNA sequencing reads is one application of next-generation sequencing (NGS) that has held a central role in biological discovery in both model and non-model organisms, with and without whole genome sequence references. A major limitation in effective building of transcriptome references is no longer the sequencing data generation itself, but the computing infrastructure and expertise needed to assemble, analyze and manage the data. Here we describe a currently available resource dedicated to achieving such goals, and its use for extensive RNA assembly of up to 1.3 billion reads representing the massive transcriptome of loblolly pine, using four major assembly software installations. The Mason cluster, an XSEDE second tier resource at Indiana University, provides the necessary fast CPU cycles, large memory, and high I/O throughput for conducting large-scale genomics research. The National Center for Genome Analysis Support, or NCGAS, provides technical support in using HPC systems, bioinformatic support for determining the appropriate method to analyze a given dataset, and practical assistance in running computations. We demonstrate that a sufficient supercomputing resource and good workflow design are elements that are essential to large eukaryotic genomics and transcriptomics projects such as the complex transcriptome of loblolly pine, gene expression data that inform annotation and functional interpretation of the largest genome sequence reference to date.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: distribution, maintenance, and enhancement—*performance analysis;* J.3 [**Life and Medical Sciences**]: biology and genetics; D.2.4 [**Distributed Systems**]: distributed applications; H.3.4 [**Systems and Software**]: distributed systems, performance evaluation (efficiency and effectiveness).

General Terms

Algorithms, Management, Measurement, Performance, Experimentation.

Keywords

NCGAS, Bioinformatics, Genome Analysis, Transcriptome, Sequence Assembly, Pine, Conifer, large memory, HPC.

XSEDE '15, July 26 - 30, 2015, St. Louis, MO, USA
 ISBN 978-1-4503-3720-5/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2792745.2792748

1. Introduction

Use of genomic sequence information for research studies across biological disciplines is increasing dramatically in both breadth and depth. Gene function and variation in contexts of disease, adaptation, growth and development, genetics and evolution are being addressed more and more with the knowledge of whole genomic information, including complete gene families and gene space sequence for structural and regulatory studies. Refined evolutionary studies have exploded further with larger numbers of publicly available sequence references, including whole genome sequence assemblies of increasingly higher and higher qualities for meaningful comparisons. Beyond the genome assembly itself, substantial contributors to quality in gene definition within genome references are transcriptome datasets. Current sequencing technologies, predominantly massively parallel, short-read "next-generation" sequencing, or NGS, provide depth of information on an unprecedented scale. RNA transcribed from the DNA code of the cell is captured and sequenced as pieces that are smaller than native length, requiring computational assembly after sequencing. The giga-scale throughputs of fragment sequencing on current instruments enable these partial transcripts to be sequenced with a high degree of redundancy. With greater depth comes greater confidence in completeness and accuracy in transcript sequence assemblies. Secondly, increasing read depth extends our ability to use

sequencing data for quantitative comparisons, such as studies of differential gene expression among multiple sample states.

Assembly of RNA sequence reads *de novo*, that is without a use of a pre-existing genomic or other transcriptome reference, is critical for discovery of polymorphic gene products and processed forms of transcripts, in addition to more readily identifiable reference-encoded transcripts. In the case of many non-model organisms such as humans and trees, *de novo* transcript assembly often builds sequence references of expressed alleles and copy number gene variants that would otherwise not be detectable in a genomic reference assembly.

Recently, genomic research innovators have taken up the challenge to build sequence references for some of the largest genomes on the planet, those of conifers, each over 20 Gb [1, 2] or roughly seven times the size of the human genome. These tree genomes with ancient origins show high levels of repeat retention from transposable element activity, high rates of heterozygosity and natural hybridization, and other factors that pose substantial impediments to accuracy and speed of genome assembly. Functional annotation of genome assemblies is further challenged by heterozygosity and intraspecies variation when for practical reasons RNA sampling must come from individuals other than the genome reference tree, and in cases where scaffolds in the genome assembly remain shorter than whole genic islands.

Loblolly pine is native to the United States, and holds economic distinction as the most commercially important tree in the Southeast [3, 4]. The reference genome for loblolly pine [5, 6] is now in widespread research use and continuing to be improved and detailed with additions from genetic mapping, transcript sequencing, additional sequencing and assembly, and comparative studies.

Here we describe the cyberinfrastructure resources and processes used to assemble and progressively analyze loblolly pine transcriptomes, references that are huge both in the number of gene expression loci they represent as well with respect to polymorphism in transcript sequence and structure.

2. Cyberinfrastructure Resources Used

Cyberinfrastructure "consists of computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked together by software and high performance networks to improve research productivity and enable breakthroughs not otherwise possible" [7].

In this paper, we discuss computational systems, people, and software in particular as the essential elements of the cyberinfrastructure that facilitated the loblolly pine assembly.

2.1 *De novo* RNA Assembly Software

Current NGS technologies generate short reads—commonly 50 to 250 bases in length—that must be assembled in order to characterize the complete set of gene transcripts (messenger RNA). RNA assembly refers to a computational algorithm and process that aligns and merges these short fragments of much longer transcripts in order to characterize the original transcript.

There are two major types of RNA assemblers: One uses a reference genome to guide assembly, while the other generates *de novo* assemblies. Reference guided assemblers require an existing reference genome and assemble reads against this backbone. In contrast, *de novo* assemblers build longer sequences from scratch, without pre-existing references.

In this paper, we use four popular *de novo de Bruijn* graph-based assemblers: Trinity [8], Velvet-Oases [9], TransABySS [10], and SOAPdenovoTrans [11]. 2013 versions were used in all cases. Briefly, Trinity is a specialized RNA assembler developed to tackle the unique challenges posed by the assembly of RNA-seq data. Oases, TransABySS, and SOAPdenovoTrans are derived from the corresponding DNA assemblers Velvet, ABySS, and SOAPdenovo.

One characteristic shared by these assemblers is that they are both CPU and memory intensive, and the amount of memory, the run time (CPU cycles), and storage space they require goes up with the size of the genome. In the worst scenario, these requirements will grow exponentially.

2.2 Mason Cluster at Indiana University

To deal with the large loblolly pine RNA-seq dataset, a personal laptop, workstation, or even a small cluster doesn't provide enough computing capacity in terms of the CPU cycles, *I/O* throughput, or, especially, the memory usage; a large supercomputing resource is a necessity for conducting this study.

The Mason cluster [12], operated by the Indiana University Pervasive Technology Institute, is a large memory cluster configured to support data-intensive, high-performance computing tasks for researchers using genome assembly software. Mason's key feature is nodes with a half terabyte of memory. This large memory profile is particularly suitable for assembly of data from next-generation sequencers, large-scale phylogenetic software, or other genome analysis applications that require large amounts of memory. Mason consists of 18 Hewlett-Packard (HP) DL580 servers, each containing four Intel Xeon L7555 eight-core processors and 512 GB of RAM, and two HP DL360 login nodes, each containing two Intel Xeon E5-2600 processors and 24 GB of RAM. The total RAM in the system is 9 TB. Each server chassis has a 10-gigabit Ethernet connection to other research systems at Indiana University and the Extreme Science and Engineering Discovery Environment (XSEDE) [13] network. More information about Mason is available at [14].

Mason is a Level 2 XSEDE Service Provider resource. It is funded entirely by IU, in support the national community of genome scientists and the mission of the National Center for Genome Analysis Support (see below). Mason is available for use by the national research community via the XSEDE Resource Allocation Committee (XRAC); researchers can apply for time on Mason through the normal XSEDE allocation application process, which has proved extremely helpful to IU. The availability of Mason is advertised to the national community more effectively through the combination of XSEDE and IU efforts. We leverage the XSEDE application process in vetting account requests. This has made it feasible for IU to make use of a high-quality allocations review process. Mason adds to XSEDE as well: It is one of the few large memory resources available in the XD ecosystem (the suite of services providers supported by XSEDE). During this research, the only other large memory system allocated by XSEDE processes was the Blacklight system at the Pittsburgh Supercomputing Center [15]. Mason is also—through collaboration with the National Center for Genome Analysis Support (see section 2.3 below)—the 'reference installation' for the suite of genome analysis software installed on the XD ecosystem and included in the XSEDE-Compatible Basic Cluster build [16].

2.3 National Center for Genome Analysis Support

The National Center for Genome Analysis Support (NCGAS) [17] is the "people" cyberinfrastructure supporting the assembly of the loblolly pine sequencing data. Led by the Indiana University Pervasive Technology Institute, NCGAS is a collaboration that includes the Texas Advanced Computing Center (TACC) [18], San Diego Supercomputing Center (SDSC) [19], and Pittsburgh Supercomputing Center (PSC) [20]. NCGAS is now sustained through funding from NSF, NIH, USDA, and the participating institutions. NCGAS was established in response to community-based concerns that the needs of biologists, and genome assembly support in particular, were not well met by the available national cyberinfrastructure resources.

In order to carry out its mission, NCGAS provides technical support in using HPC systems, bioinformatic support in determining the appropriate method for a given dataset, and actual assistance in running computations. NCGAS particularly supports the analysis of next-generation sequencer data in *de novo* assembly where no reference genome is available, such as transcriptomic studies, and metagenomic projects where the combined genome in an environmental sample is simultaneously sequenced.

By coordinating efforts between multiple supercomputing centers, NCGAS is creating a service-oriented infrastructure that is designed to deliver—and assist biologists in the use of—supercomputer clusters for genome analysis, including many supercomputers and supercomputer clusters that are part of XSEDE [21]. NCGAS, for example, has created several optimizations of the Trinity RNA-seq software [22] and contributed them back to the community code base.

In this study, NCGAS began operations shortly after the PineRefSeq project launched in February 2011, and almost immediately became involved in managing the massive pine NGS datasets. NCGAS has contributed essential computing, data management, and bioinformatic expertise to building the first comprehensive conifer gene expression reference, the loblolly pine transcriptome.

3. MATERIALS AND METHOD

3.1 RNA-seq Sampling and Sequencing

The sequence datasets used in this paper are constructed with RNA-seq technology. Unlike traditional approaches like micro-array, RNA-seq technology converts mRNA to cDNA and uses it as input to next-generation sequencing. Compared to preexisting technologies, RNA-seq technology provides very high throughput [23]. For this paper, Illumina Hi-Seq sequencer [24] datasets of indexed paired-reads of 100nts were used. RNA-seq libraries included a variety of tissues, developmental stages, and environmental stress conditions (Fuentes-Soriano *et al.*, in preparation).

3.2 Data Preprocessing

To cope with problems of sequence-adapter contamination and low quality reads, the raw RNA-seq reads are preprocessed before further analysis. Here, FastQC [25] and Trimmomatic [26] are used to remove over-represented sequences and low quality reads from our initial RNA-seq datasets. Reads for each sample are trimmed and screened to remove ends of biased nucleotide representation, as well as reads with quality values lower than 30, where a quality value (Q-value) 30 indicates a 0.1% probability of an incorrect base call. After trimming and quality filtering, we had a total of 50 Gbases (about 1.7M read pairs) of short read data from 27 sample sets.

In addition, digital normalization [27] is applied to all samples after trimming and quality filtering, to create an alternative collection of datasets. The digital normalization mentioned here is a single-pass computational algorithm to reduce sampling variation, remove errors, and scale the *de novo* assembly of the transcriptome.

3.3 RNA Assembly

We use the trimmed and filtered sequence reads from pre-processing (section 3.2) as input for *de novo* RNA assembly with four popular assemblers: Trinity, Velvet-Oases, TransABySS, and SOAPdenovoTrans (as described in section 2.1). 2013 versions were used for each.

All assemblies are performed on a single node of the Mason cluster, with 32 processors and 500 GB of memory. In addition, six different k-mer sizes (31, 41, 51, 61, 71, and 81) are used for Velvet-Oases, TransABySS, and SOAPdenovoTrans assemblies; Trinity is only run with its default k-mer value of 25.

More than 600 RNA assemblies were generated using each possible combination of assembly parameters, including sequencing samples, type of assemblers, +/- normalization, and k-mer values. Downstream comparative analyses not described here classified these according to sequence redundancy, protein coding structure, and other factors before a final set of assemblies was selected for the transcriptome reference

3.4 Compute Settings

To take advantage of the Mason cluster's power, we introduce steps to increase workflow efficiencies. First, while the GNU compiler is a good default choice, other compilers—such as the Intel or PGI compilers—can offer performance advantages. Since the Mason cluster is based on the Intel Nehalem processor, when possible we used Intel compilers to build all packages.

Second, for tasks using single thread implementations (such as trimming, filtering, and redundancy removing), we use a customized python script to partition the input dataset into a collection of small subsets and then map multiple tasks for small subsets in parallel. Once all partitioned jobs are done, we merge the output from each run, based on the initial splitting. For tasks using multithreading (such as *de novo* assembly and mapping), we use the maximum number of threads on all the available processors.

Third, several checkpoints are introduced to the analysis pipelines to prevent our long-running jobs from restarting when jobs run into a failure due to either software or hardware issues. We also apply a job array approach, one of the high performance computing techniques, for distributing similar tasks to multiple computing resources in parallel, such as constructing assemblies from multiple k-mer values. In this way, a single job submission can spawn multiple jobs simultaneously.

Finally, when running applications on cluster-wide file systems, the *I/O* bandwidth available to the job depends heavily on the *I/O* characteristics of other concurrent jobs. But when the running directory of a job is located on a node local file system, this dependency is removed and there will be less fluctuation in execution time. On the Mason cluster, the memory file system

/dev/shm is available on all nodes, offering up to 256 GB of very fast local scratch storage. Therefore, when possible, we place input and output files on the memory file system, greatly improving the performance of *I/O* intensive applications.

4. DISCUSSION

4.1 Assembler Comparison

How many computing resources, such as CPU cycles and memory usage, are needed to complete an assembly job is a substantial and practical issue to ponder when selecting assembly software—especially when dealing with very large datasets.

In order to compare the requirements of each assembly application, we compute the average memory and run time used for each sample by the different *de novo* assemblers, as shown in Table 1. The samples are all paired-end reads, and the average sample size is around 3.1 billion bps. Overall, Trinity requires the most resources, in both memory and CPU cycles, whereas TransABySS consumes the least amount of memory—and SOAPTrans takes the shortest time to run.

Table 1. Average memory and run time used for a RNA assembly task, per sample reads by different assemblers.

Assembler	k-mer values	Memory Used	Run Time
TransABySS	31,41,51,61,71,81	14 GB	6.5 hrs
Trinity	25 (default)	300 GB	8.0 hrs
SOAPTrans	31,41,51,61,71,81	35 GB	2.5 hrs
Velve/Oases	31,41,51,61,71,81	50 GB	8.0 hrs

Comparing the demand for resources among assemblers gives us an idea of when memory and CPU cycles will be an impediment to assembly jobs. This information can also help us select an appropriate computing facility for running assembly jobs, as well as plan a reasonable schedule to carry out the entire analysis workflow.

The results shown in Table 1 also indicate the necessity of the large memory Mason cluster for this study. When assembling the pine RNA-seq data, most of the assemblers require more than 32 GB of memory (Trinity needs up to 300 GB), and the Mason cluster is one of the two XSEDE computing resources that can meet this demand.

4.2 Computing Resource Use

4.2.1 CPU cycles

The major computing jobs of this study were executed between 2012 and 2014, lasting 21 months. How many jobs and CPU core hours have been dedicated to these computing jobs? We collected the number of jobs we ran and determined the core hours for each job.

As shown in Figure 1, over the course of this study 1,221 major computing jobs, which required more than one hour of run time each, were conducted. The result is that there were roughly 60 jobs per month—or two jobs per day, every day during the major computing period of this study. This enormous amount of traffic indicates how this research project benefited from the national supercomputing cyberinfrastructure.

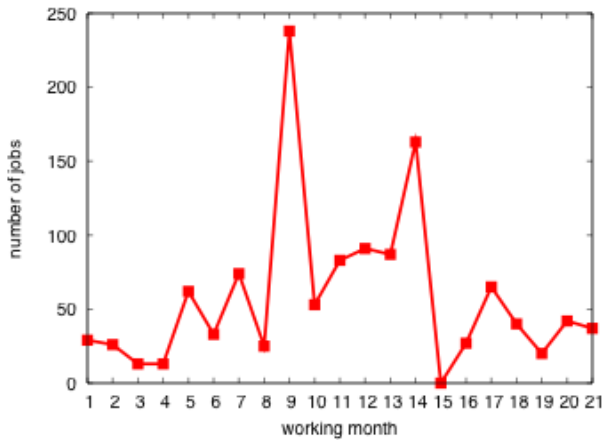

Figure 1. Number of jobs per month across the 21-month period of major computing works

CPU core hours is another indication of how intensive a computing regime is. Figure 2 shows the accumulated CPU core hours per working month for all the computing jobs we executed on Mason.

In total, 97,274.23 CPU hours were used by the loblolly pine project. On a single 8-core workstation, this work would take up to 506 days. Clearly, national cyberinfrastructure is vital to these endeavors, as well as NCGAS's role in advancing and accelerating scientific research.

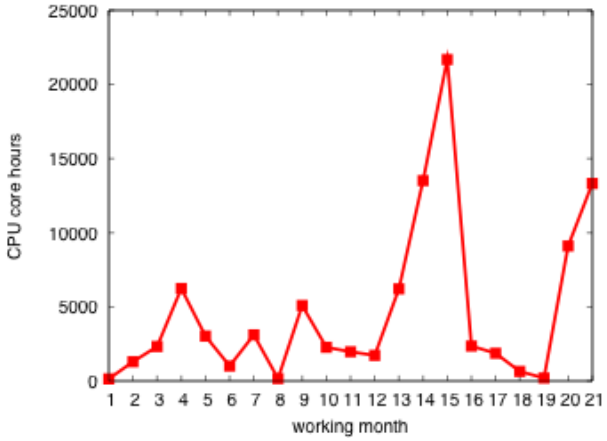

Figure 2. Accumulated CPU core hours per month across the 21 months of major work.

4.2.2 Memory Usage

Any computer software needs a certain amount of computing power to work; memory requirements can especially limit a program's efficiency when dealing with large datasets.

Figure 3 shows the frequency distribution of the maximum memory usage per job over the course of this project's analysis period: Roughly half of the analysis jobs we ran for this study required more than 16 GB of memory. More importantly, 26% of our computing jobs used more than 64 GB of memory, representing the second largest group of jobs.

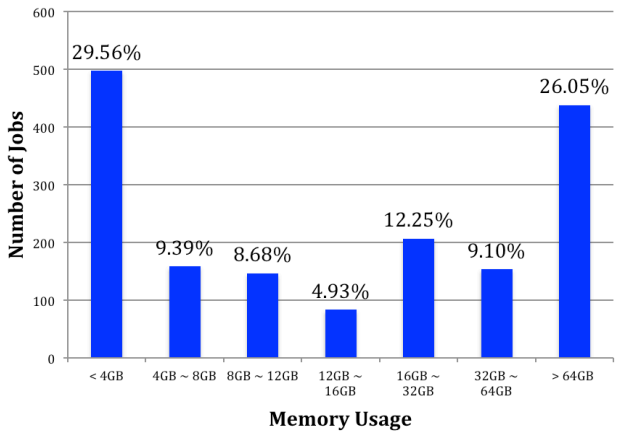

Figure 3. Histogram of the maximum memory usage per job. The number shown atop each bar represents the percentage of total jobs in this group.

Small memory jobs usually come from trimming, filtering, normalization, data partitioning, data transfering, and other data preparing tasks. Large memeory jobs are mainly the actual assembly pipelines. The memory usage pattern shown here reveals that, although small jobs are still the majority, one third of our computing jobs are over the capacity of traditional clusters.

5. CONCLUSION

In this paper, we describe a workflow to perform extensive RNA assembly with inputs of up to 1.3 billion reads, using four different software installations. The resulting assembly was used to build the first comprehensive conifer gene expression reference. These transcript sequences are being used on their own for the individual gene expression and sequence variation information they provide, and most significantly as annotation evidence for the loblolly pine genome reference. The currently available genome assembly spans 23.2 Gbp and contains 20.1 Gbp with an N50 scaffold size of ≥ 66.9 kbp [28] making it both the highest quality conifer genome reference and the largest genome reference built to date. Transcriptome references built using the XSEDE resource described here have contributed essential evidence allowing the annotation of transcribed regions and gene functions. Genome sequencing technologies create more sequence data than ever before. Therefore, biologists often require high performance cyberinfrastructure for their analysis. In addition, a good workflow design is essential to time efficiencies. The Mason cluster, an XSEDE second tier resource at Indiana University, provides the necessary fast CPU cycles, large memory, and high I/O throughput for conducting research on the scale of large eukaryotic genomics.

The mission of NCGAS is to provide support to biologists, including delivery and assistance in the use of supercomputer clusters for genome analysis, including many supercomputers and supercomputer clusters that are part of XSEDE. In addition to facility hardware enabling important contributions to the first U. S. conifer reference project and other large eukaryotic genomics projects, NCGAS has contributed high performance computing, data management and bioinformatics expertise as an integrated package that is essential to the initiation and success of such projects.

6. ACKNOWLEDGMENTS

This work was supported in part by USDA NIFA grant 2011-67009-30030, PineRefSeq, led by the University of California, Davis to Keithanne Mockaitis; and NCGAS funded by NSF under award No. 1062432. Craig Stewart is PI on the NCGAS award; William K. Barnett is NCGAS director, and Thomas Doak is NCGAS manager. The Indiana University Pervasive Technology Institute was established with generous funding from the Lilly Endowment, Inc. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Lilly Endowment, US Department of Agriculture, or National Science Foundation.

7. REFERENCES

[1] Birol, I., et al., *Assembling the 20 Gb white spruce (Picea glauca) genome from whole-genome shotgun sequencing data.* Bioinformatics, 2013. **29**(12): p. 1492--1497.

[2] Nystedt, B., et al., *The Norway spruce genome sequence and conifer genome evolution.* Nature, 2013. **497**(7451): p. 579--584.

[3] Nix, S. *Ten Most Common Trees in the United States. About.com Forestry*. [cited 2015 4 Apr]; Available from: http://forestry.about.com/b/2012/07/21/ten-most-common-trees-in-the-united-states.htm.

[4] University, N.C.S. *Tree Improvement Center*. 2015 [cited 2015 4 Apr]; Available from: http://www.treeimprovement.org/public/about/species-interest/loblolly-pine/loblolly-pine.

[5] Neale, D.B., et al., *Decoding the massive genome of loblolly pine using haploid DNA and novel assembly strategies.* Genome Biol, 2014. **15**(3): p. R59. DOI: 10.1186/gb-2014-15-3-r59.

[6] Wegrzyn, J.L., et al., *Unique features of the loblolly pine (Pinus taeda L.) megagenome revealed through sequence annotation.* Genetics, 2014. **196**(3): p. 891-909. DOI: 10.1534/genetics.113.159996.

[7] Stewart, C.A., et al., *What is cyberinfrastructure*, in *Proceedings of the 38th annual ACM SIGUCCS fall conference: navigation and discovery*. 2010, ACM: Norfolk, Virginia, USA. p. 37-44. DOI: 10.1145/1878335.1878347.

[8] Grabherr, M.G., et al., *Full-length transcriptome assembly from RNA-Seq data without a reference genome.* Nat. Biotechnol., 2011. **29**(7): p. 644--652.

[9] Schulz, M.H., et al., *Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels.* Bioinformatics, 2012. **28**(8): p. 1086--1092.

[10] Robertson, G., et al., *De novo assembly and analysis of RNA-seq data.* Nat. Methods, 2010. **7**(11): p. 909--912.

[11] Li, R., et al., *SOAP2: an improved ultrafast tool for short read alignment.* Bioinformatics, 2009. **25**(15): p. 1966--1967.

[12] *Indiana University Mason Cluster*. [cited 2015 4 Apr]; Available from: https://kb.iu.edu/d/bbhh/.

[13] *Extreme Science and Engineering Discovery Environment*. [cited 2015 4 Apr]; Available from: https://http://www.xsede.org.

[14] *Getting started on Mason*. [cited 2015 4 Apr]; Available from: https://kb.iu.edu/d/beyh.

[15] *PSC Blacklight User Guide. XSEDE*. [cited 2015 4 Apr]; Available from: https://portal.xsede.org/psc-blacklight.

[16] Fischer, J., et al., *Methods For Creating XSEDE Compatible Clusters*, in *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*. 2014, ACM: Atlanta, GA, USA. p. 1-5. DOI: 10.1145/2616498.2616578.

[17] *National Center for Genome Analysis Support*. Available from: http://www.ncgas.org/.

[18] *Texas Advanced Computing Center*. [cited 2015 4 Apr]; Available from: http://www.tacc.utexas.edu/.

[19] *San Diego Supercomputer Center*. [cited 2015 4 Apr]; Available from: http://www.sdsc.edu/.

[20] *Pittsburgh Supercomputing Center*. [cited 2015 4 Apr]; Available from: http://www.psc.edu/.

[21] LeDuc, R.D., et al., *National Center for Genome Analysis support leverages XSEDE to support life science research*, in *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. 2013, ACM: San Diego, California, USA. p. 1-7. DOI: 10.1145/2484762.2484790.

[22] Henschel, R., et al., *Trinity RNA-Seq assembler performance optimization*, in *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond*. 2012, ACM: New York, NY, USA. p. 45:1--45:8. DOI: 10.1145/2335755.2335842.

[23] Wang, Z., M. Gerstein, and M. Snyder, *RNA-Seq: a revolutionary tool for transcriptomics.* Nat. Rev. Genet., 2009. **10**(1): p. 57--63.

[24] *Illumina Hi-Seq sequencer*. [cited 2015 4 Apr]; Available from: http://www.illumina.com/systems/hiseq_2500_1500.html.

[25] Andrews, S., *FastQC A Quality Control tool for High Throughput Sequence Data.* http://www.bioinformatics.babraham.ac.uk/projects/fastqc/, 2015. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

[26] Bolger, A.M., M. Lohse, and B. Usadel, *Trimmomatic: a flexible trimmer for Illumina sequence data.* Bioinformatics, 2014. **30**(15): p. 2114--2120. DOI: 10.1093/bioinformatics/btu170.

[27] Brown, C.T., et al., *A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data*. 2012. http://arxiv.org/abs/1203.4802.

[28] Zimin, A., et al., *Sequencing and assembly of the 22-gb loblolly pine genome.* Genetics, 2014. **196**(3): p. 875--890. DOI: 10.1534/genetics.113.159715.