Video Article

Leveraging CyVerse Resources for *De Novo* Comparative Transcriptomics of Underserved (Non-model) Organisms

Blake L. Joyce[1,2], Asher K. Haug-Baltzell[3], Jonathan P. Hulvey[4], Fiona McCarthy[5], Upendra Kumar Devisetty[1,6], Eric Lyons[1,2,3]

[1]BIO5 Institute, University of Arizona

[2]The School of Plant Sciences, University of Arizona

[3]Genetics GIDP, University of Arizona

[4]Biology Department, University of Massachusetts Amherst

[5]School of Animal and Comparative Biomedical Sciences, University of Arizona

[6]CyVerse, University of Arizona

Correspondence to: Blake L. Joyce at bjoyce3@email.arizona.edu

URL: https://www.jove.com/video/55009
DOI: doi:10.3791/55009

Keywords: Genetics, Issue 123, *de novo* transcriptome assembly, differential gene expression, non-model organisms, CyVerse, cyberinfrastructure, bioinformatics, cloud computing

Date Published: 5/9/2017

Citation: Joyce, B.L., Haug-Baltzell, A.K., Hulvey, J.P., McCarthy, F., Devisetty, U.K., Lyons, E. Leveraging CyVerse Resources for *De Novo* Comparative Transcriptomics of Underserved (Non-model) Organisms. *J. Vis. Exp.* (123), e55009, doi:10.3791/55009 (2017).

Abstract

This workflow allows novice researchers to leverage advanced computational resources such as cloud computing to carry out pairwise comparative transcriptomics. It also serves as a primer for biologists to develop data scientist computational skills, *e.g.* executing bash commands, visualization and management of large data sets. All command line code and further explanations of each command or step can be found on the wiki (https://wiki.cyverse.org/wiki/x/dgGtAQ). The Discovery Environment and Atmosphere platforms are connected together through the CyVerse Data Store. As such, once the initial raw sequencing data has been uploaded there is no more need to transfer large data files over an Internet connection, minimizing the amount of time needed to conduct analyses. This protocol is designed to analyze only two experimental treatments or conditions. Differential gene expression analysis is conducted through pairwise comparisons, and will not be suitable to test multiple factors. This workflow is also designed to be manual rather than automated. Each step must be executed and investigated by the user, yielding a better understanding of data and analytical outputs, and therefore better results for the user. Once complete, this protocol will yield *de novo* assembled transcriptome(s) for underserved (non-model) organisms without the need to map to previously assembled reference genomes (which are usually not available in underserved organism). These *de novo* transcriptomes are further used in pairwise differential gene expression analysis to investigate genes differing between two experimental conditions. Differentially expressed genes are then functionally annotated to understand the genetic response organisms have to experimental conditions. In total, the data derived from this protocol is used to test hypotheses about biological responses of underserved organisms.

Video Link

The video component of this article can be found at https://www.jove.com/video/55009/

Introduction

Homo sapiens and several key model animal species such as *Drosophila melanogaster*, *Mus musculus*, and *Danio rerio* represent the majority of current and past functional genomics work. However, the rapidly decreasing cost of high-throughput sequencing technology is providing opportunities for functional genomics in non-model (*a.k.a.* "neglected" or "underserved") animal species[1]. This is an important transition in genomics as non-model organisms frequently represent economically relevant species (*e.g.* oysters, shrimp, crab) and offer opportunities to investigate novel phenotypes and biological systems outside the scope of those found in model species.

Although underserved organisms present an attractive opportunity to investigate unique biological systems, several challenges face researchers particularly during bioinformatic analysis. Some of these challenges are innate to processing large data sets, while others result from the lack of genetic resources available to researchers working in underserved organisms such as a reference genome, organism specific ontologies, *etc.* The challenges of nucleic acid isolation and sequencing are often routine in comparison with those of data analysis, and as such bioinformatic analyses generally proves to be the most underestimated cost of sequencing projects[2]. For example, a basic next-generation sequencing bioinformatic analysis might consist of the following steps: quality filtering and trimming of raw sequencing reads, assembly of short reads into larger contiguous pieces, and annotation and/or comparisons to other systems to gain biological understanding. While seemingly simple, this example workflow requires specialty knowledge and computational resources beyond the scope of a lab-bench computer, placing it out of reach of many scientists studying non-model organisms.

Innate challenges can be infrastructure- or knowledge-based. A classic infrastructure challenge is access to appropriate computational resources. For example, assembly and annotation rely on computationally intensive algorithms that require powerful computers or computer clusters, having large amount of RAM (256 GB-1 TB) and several processors/cores to run. Unfortunately, many researchers either do not have access to such computing resources or do not have the knowledge needed to interact with these systems. Other researchers might have access to high-performance computing clusters through their universities or institutions, but access to these resources might be limited and sometimes results in charges per compute hour, *i.e.* the number of CPU processors multiplied by the number of real-time "clock hours" that those processors are running. Leveraging a cyberinfrastructure system funded by the US National Science Foundation such as CyVerse[3] that provides free access to compute resources for researchers, in the United States and around the world, can help alleviate infrastructure challenges, as will be demonstrated here.

An example of a typical knowledge-based challenge is understanding the software needed for complete analyses. To effectively conduct a sequencing-based project, researchers need to be familiar with the myriad of software tools that have been developed for bioinformatic analyses. Learning each package is difficult in its own right, but is exacerbated by the fact that packages are constantly being upgraded, rereleased, put together into new workflows, and sometimes become restricted for use under new licenses. In addition, linking the inputs and outputs of these tools sometimes requires transforming data types to make them compatible, adding another tool to the workflow. Finally, it is also difficult to know which software package is 'the best' for an analysis, and frequently identifying the best software for particular experimental conditions is a matter of subtle differences. In some cases, useful reviews of software are available, but due to the continuing release of new updates and software options, these rapidly go out of date.

For researchers investigating underserved organisms, these innate challenges come in addition to the challenges associated with analyzing data in a novel organism. These underserved organism-specific challenges are best illustrated during gene annotation. For example, underserved organisms frequently do not have a closely related model organism that can be reasonably used to identify gene orthology and function (*e.g.* marine invertebrates and *Drosophila*). Many bioinformatic tools also require "training" to identify structural motifs, which can be used to identify gene function. However, training data is usually only available for model organisms, and training hidden Markov models (HMMs) is outside the purview of biologists, and even many bioinformaticians. Lastly, even if annotations can be carried out using data from model organisms, some gene ontologies associated with model organisms do not make sense when the biology and natural history of the underserved organism is considered (*e.g.*, transferring information from *Drosophila* to shrimp).

In light of these challenges, bioinformatic resources need to be developed with researchers conducting *de novo* analyses on underserved organisms specifically in mind. The next several years of functional genomics sequencing projects will help to close the gap between model and underserved organisms (https://genome10k.soe.ucsc.edu/), but there are many tools that will need to be developed to address the challenges considered above. CyVerse is dedicated to creating ecosystems of interoperability by linking existing cyberinfrastructure and third party applications to deliver data management, bioinformatic analysis tools, and data visualizations to life scientists. Interoperability helps to smooth the transitions between bioinformatic applications and platforms by providing scalable computing resources, and limiting file format conversions and the amount of data transferred between platforms. CyVerse offers several platforms, including the Discovery Environment (DE[4], Atmosphere[5], and the Data Store[3]. The DE is web-based and has many common bioinformatics analytic tools converted to user-friendly point-and-click formats (called "apps"), and is the graphical user interface (GUI) for the Data Store where large data sets (*i.e.* raw sequencing reads, assembled genomes) are stored and managed. Atmosphere is a cloud computing service that offers researchers increased flexibility for using Virtual Machine computational resources, which have a wide range of bioinformatics tools pre-installed. Both of these platforms are linked to the Data Store, and can be used together to create workflows such as that described here. This report focuses on a *de novo* transcriptome assembly and differential gene expression analysis workflows, and further addresses some best practices associated with developing and conducting bioinformatic analyses. An explanation of the broader mission of CyVerse (http://www.cyverse.org/about) and detailed platform descriptions (http://www.cyverse.org/learning-center) are publicly available. All analyses described herein use the Discovery Environment[4] (DE) and Atmosphere[5], and are presented in a manner to make them accessible to researchers of all computational levels. DE workflows and Atmosphere images can be referenced directly using URLs to ensure long-term provenance, reusability, and reproducibility.

Protocol

NOTE: The overall protocol has been numbered according to folders that will be created and named in step 1.2 (**Figure 1** and **2**). This protocol represents a standard comparative *de novo* transcriptome analysis, and every step detailed here may not be necessary for all researchers. This workflow is documented thoroughly on a companion tutorial wiki, which also contains all additional files and links to documents of interest 3rd party developers for each analysis package (**Table 1**). Links to this material will be included throughout this protocol for easy access to this information. Best practices are notes provided to users as suggestions for the best way to accomplish tasks or for users to consider, and will be communicated through notes in the protocol. A folder of example data input and analytical output is publicly available to users, and is organized as suggested in the protocol (*de novo* transcriptome assembly and analysis.

1. Set up the Project, Upload Raw Sequencing Reads, and Assess Reads Using FastQC

1. Get access to Atmosphere and the Discovery Environment.
    1. Request a free CyVerse account by navigating to the registration page (*e.g.* person@institution.edu).
    2. Fill in the required information and submit.
    3. Navigate to the main webpage (http://www.cyverse.org/), and select "Sign In" at the top toolbar. Select "Cyverse Login" and sign in using your CyVerse credentials.
    4. Navigate to the Apps & Services tab, and request access to Atmosphere. Access to the Discovery Environment is automatically granted.
2. Set up the project and move data to the Data Store.
    1. Log into the Discovery Environment (https://de.iplantcollaborative.org/de). Select the “Data” tab to bring up a menu containing all the folders in the Data Store.

2. Create a main project folder that will house all of the data associated with the project. Find the toolbar at the top of the data window and select File | New Folder. Do not use spaces or special characters in the folder names or any input/output file names *e.g.* "!@#()[]{}:;$%^&*." Instead, use underscores or dashes, *i.e.* "_" or "-" where appropriate.
3. Create five folders within the main project folder to organize analyses (**Figure 1**) Name the folders as follows without commas or quotation marks: "1_Raw_Sequence," "2_High_Quality_Sequence," "3_Assembly," "4_Differential_Expression," "5_Annotated_Assembly." Subfolders will be placed into each of these main project folders (**Figure 2**).

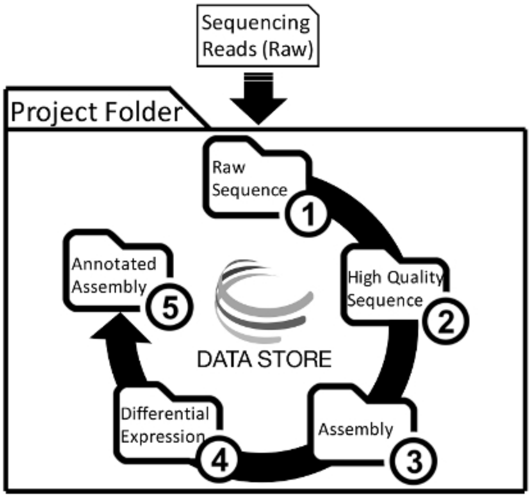

Figure 1: A General Overview of Project Folder Organization and the *De Novo* Transcriptome Assembly and Analysis Workflow. Users will upload raw sequencing reads into the main project folder on the Data Store, and then place the results from each step into separate folders. Please click here to view a larger version of this figure.

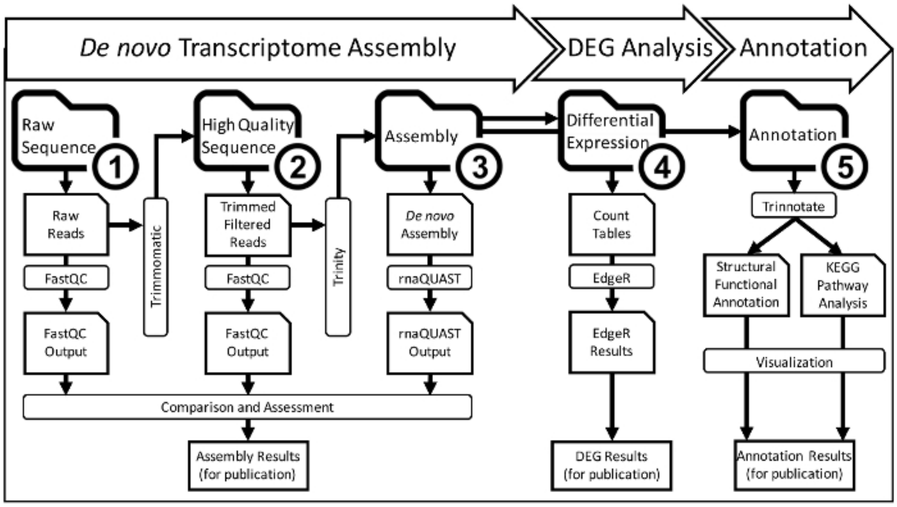

Figure 2: A Detailed Overview of the *De Novo* Transcriptome Assembly and Analysis Workflow that Occurs within CyVerse Cyberinfrastructure. The entire assembly and analysis workflow will be completed in five steps which each get their own folder (bolded, numbered folder icons). Each of the five numbered workflow step folders has subfolders containing output data from bioinformatic analyses (folder icons). Inputs for analysis come from one subfolder and then move into another folder through the output of an analysis program (rectangle boxes). The final data from the first three steps is compared and prepared for publication. Ultimately, this scheme yields a main project folder that has stepwise analysis for collaborators and/or manuscript reviewers can quickly understand the workflow and repeat it using each file if necessary. Please click here to view a larger version of this figure.

3. Upload raw FASTQ sequence files into the folder "1_Raw_Sequence" into a subfolder entitled "A_Raw_Reads" using one of the following three methods.
   1. Use the Data Store simple upload feature to navigate to the Data window toolbar by clicking on the data button in the main DE desktop, and select Upload | Simple Upload from Desktop. Select the Browse button to navigate to the raw FASTQ sequencing files on the local computer. This method is only suitable for files under 2 GB.
   2. Select the Upload button at the bottom of the screen to submit the upload. A notification will register in the top right of the DE in the bell icon that the upload has been submitted. Another notification will register when the upload is complete.
   3. Alternatively, use Cyberduck to transfer larger files (https://wiki.cyverse.org/wiki/x/pYcVAQ). Install Cyberduck and then run as a program on the local computer's desktop.

    4. Lastly, download iCommands and install onto the local computer according to instructions (https://wiki.cyverse.org/wiki/display/DS/Using+iCommands).

4. Assess uploaded, raw sequencing reads using the FastQC app in the DE.
    1. Select the "Apps" button on the main DE desktop to open a window containing all of the analysis apps available in the DE.
    2. Search and open the window for the FastQC tool in the search toolbar at the top of the window. Open the multi-file version if there is more than one FASTQ file. Select File | New Folder to create a folder named "B_FastQC_Raw_Reads" and select this folder as the output folder.
    3. Load the FASTQ read files into the tool window called "Select input data" and select "Launch Analysis."
    4. Open the .html or .pdf file to view the results once the analysis is complete. FastQC runs several analyses that test different aspects of the read files (**Figure 3**).

2. Trim and Quality Filter Raw Reads to Yield High Quality Sequence

Note: Use either the Trimmomatic app or the Sickle app.

1. Search for the programmable Trimmomatic app in the DE and open it as before.
    1. Upload the folder of raw FASTQ read files into the "Settings" section.
    2. Select whether the sequencing files are single- or paired-end.
    3. Use the standard control file provided by selecting the Browse button and pasting /iplant/home/shared/Trinity_transdecoder_trinotate_databases into the "Viewing:" box. Select the file named Trimmomaticv0.33_control_file and launch the analysis. The file can be downloaded, the settings edited, and then uploaded into the second project folder to create a custom trimming script.
    4. Optional: If the FastQC analysis identified adapter sequences, use the ILLUMINACLIP setting to trim Illumina adapters. Select the appropriate adapter file in the folder /iplant/home/shared/Trinity_transdecoder_trinotate_databases as above.

2. Quality trimming sequence reads using Sickle.
    1. Search and open the Sickle app in the DE. Select the trimmed FASTQ reads as input reads, and rename output files. Include quality settings in the options. Typical settings are Quality format: illumina, sanger, solexa; Quality threshold: 20; Minimum length: 50.
    2. Move all output into the trimmed and filtered folder (2_High_Quality_Sequence).

3. Assess the final reads using FastQC and compare to previous FastQC reports. Select the .html file to bring up a webpage of all results. Select the folder of image files (.png) that are provided in the output if that cannot be viewed.

3. *De Novo* Transcriptome Assembly Using Trinity in Atmosphere

1. Open the most current version of the Atmosphere instance by navigating to the wiki page (https://wiki.cyverse.org/wiki/x/dgGtAQ). Select the link for the most recent version of the Trinity and Trinotate image. Alternatively, search “Trinotate” in the Atmosphere image search tool (https://atmo.iplantcollaborative.org/application/images) to bring up all versions of the Trinity and Trinotate images.
    1. Select the "Log in to launch" button and then name the Atmosphere instance.
    2. Select an instance size of either "medium3" (CPU: 4, Mem: 32GB) or "large3" (CPU: 8, Mem: 64 GB). Launch the instance, and wait for it to build. In some rare cases CyVerse undergoes maintenance to update platforms. Existing instances are available during these updates, but it may not be possible to create new instances. Visit the CyVerse Status page to see the current state of any platform (http://status.cyverse.org/).

2. Open the instance once it is ready by clicking on the name and then selecting "Remote Desktop" on the bottom of the menu on the right. Allow Java and VNC Viewer if asked. Select the "Connect" button in the VNC Viewer window, and then select "Continue."
    1. Log in to open a separate window that will be the new cloud computing instance.
    2. Move the trimmed and/or filtered FASTQ read files into the instance using one of the three methods described in steps 1.3.1 - 1.3.4. Use the Internet browser to access the DE and download files just as before on the local computer. Or use iCommands installed on these images to quickly transfer large data sets.

3. Running Trinity to assemble high quality reads.
    1. Set up the analysis folder on the Atmosphere instance. Use the script available in the DE (/iplant/home/shared/Trinity_transdecoder_trinotate_databases) or copy and paste the commands from wiki page (https://wiki.cyverse.org/wiki/x/dgGtAQ). Explanation of all commands can be found on the wiki page.
    2. Once the analysis folder and the Trinotate databases are established, run the Trinity assembler using the commands from above. There are several output files, but the most important is the final assembly file entitled "Trinity.fasta." Rename this FASTA file to be unique to the organism and treatment of the assembled reads before moving it into the Data Store (folder 3_Assembly) to minimize potential confusion.
    NOTE: Output counts tables for differential gene expression analysis into a folder (4_Differential_Expression).

4. Assess the assembly using rnaQUAST (**Figure 4**).
    1. Move the Trinity output files into the folder "3_Assembly" in the DE and label the folder "A_Trinity_de_novo_assembly." Give each transcriptome that was assembled a subfolder inside the "A_Trinity_de_novo_assembly" folder with unique names including the scientific name of organisms and treatments associated with each transcriptome. Create another subfolder called "B_rnaQUAST_Output" in the "3_Assembly folder."
    2. Open the app titled "rnaQUAST 1.2.0 (denovo based)" and name the analysis and select "B_rnaQUAST_Output" as the output folder.

        1. Add the *de novo* assembly FASTA file(s) to the "Data Input" section. In the "Data Output" section, type a unique name for the *de novo* assembly. This will create a folder of rnaQUAST output files inside of the folder "B_rnaQUAST_Output."

    3. Select additional options in the "GenemarkS-T Gene Prediction," "BUSCO," and "Parameters" sections.
        1. Select prokaryote in the "GenemarkS-T Gene Prediction" section if the organism is not eukaryotic.
        2. Run BUSCO to select the browse button and copy the path iplant/home/shared/iplantcollaborative/example_data/BUSCO.sample.data into the "Viewing:" box and press enter. Select the most specific BUSCO folder that is available for the organism.
        NOTE: BUSCO will assess the assembly for lineage-specific core genes, and output what percentage of core genes is found. There are general folders, *e.g.* eukaryote, and more specific lineages, *e.g.* arthropoda.

5. Search for "Transcript decoder" and run Transdecoder on the *de novo* Trinity assembly output FASTA file in the Discovery Environment.
6. Move the output .pep file into the *de novo* assembly (3_Assembly) folder for use in step 5 Annotation.

4. Pairwise Differential Expression Using DESeq2 in the DE

1. Open the DESeq2 app in the DE as described previously. Name the analysis and select the output folder as 4_Differential_Expression.
2. In the "Inputs" section, select the counts table file from the Trinity assembly run and the column that the contig names can be found in that counts table.
3. Input the column headers from the counts data table file to determine which columns are compared. Include the commas between each of the conditions. Do not include the first column header that contains the contig names.
4. For replicates, repeat the same name (*e.g.*, Treatment1rep1, Treatment1rep2, Treatment1rep3 would become Treatment1, Treatment1, Treatment1). In the second line, provide the names of the two conditions to be compared (*e.g.*, Treatment1, Treatment2). Match the column header names provided in the first line.
NOTE: These column headers must be alphanumeric and cannot contain any special characters.

5. Annotation Using Trinotate

1. Run each part of Trinotate in the Atmosphere cloud computing instance. Note: Bash commands are provided in a txt file to be copied, pasted, and then modified before running on the DE (/iplant/home/shared/Trinity_transdecoder_trinotate_databases) or on the wiki page (https://wiki.cyverse.org/wiki/x/dgGtAQ). If annotating multiple assemblies, annotate each assembly one at a time and then transfer completed annotations files back to folder “5_Annotation” each with a unique folder corresponding with the assembly name.
    1. Run the bash command for searching Trinity transcripts. Change the number of threads to match how many CPUs are on the instance, *i.e.* medium has 4 CPUs and large has 8 CPUs. Refer to step 3.1.2 for more details. Change the command Trinity.fasta to match the assembly FASTA file name.
    NOTE: BLAST+ searches will require the most time. It may be days before it completes. The cloud computer activity can be checked in Atmosphere without having to bring up the VNC Viewer.
    2. Run the bash command for searching Transdecoder-predicted proteins. As before, change the threads number and file name to match the conditions in 5.2.1.
    3. Run the bash command for HMMER and change the number of threads as above.
    4. Run the bash command for signalP and tmHMM if needed. SignalP will predict signal peptides and tmHMM predicts transmembrane protein motifs.

2. Loading results into the SQLite database
    1. Once all of the above analyses are completed, run the bash command to load output files into a final SQLite annotation database. Remove any commands for analyses that were not run.
    2. Export the SQLite database into a .xls file for viewing in popular table viewers.

Representative Results

Once the project organization files have been created (**Figure 1** and **2**), the first task in this workflow is to assess the raw sequencing files, and then to clean them by trimming and quality filtering. FastQC will generate human-readable summary statistics about the quality scores and length of sequences from the FASTQ file format. The FastQC figures are then compared before and after trimming to assess whether the final reads are high quality and therefore suitable for assemble. "Per base sequence quality" shows the average quality of reads across each base pair of sequencing. It is best to have a phred quality score above 20-28 indicated by the colors on the FastQC figures. "Per sequence quality score" determines whether quality filtering of reads may be necessary. If too many reads have an average score below 20-25 then it may be necessary to filter based on average read quality. "Per base sequence content" should show an even distribution across all four nucleotide bases. If there is bias in the nucleotide content is shown, then trimming ends may be necessary. "Per base GC content should also be even across all positions. If there is a wobble the reads may need to be trimmed as in 1.4.4.3. "Per sequence GC content" should be a normal distribution. Adapter or polymerase chain reaction (PCR) products can contamination in the sequencing library and skew the normal distribution. In this case, adapter trimming may be necessary. "Sequence length distribution" gives the average lengths of all reads. Reads smaller than 35-45 base pairs are usually filtered out. "Sequence duplication levels" show how many times a given read's sequence is seen within the library. Highly duplicated read sequence and count are provided in the "Overrepresented sequences" section. FastQC also attempts to identify whether the duplicated reads are adapter sequence or other known sequences associated with sequencing platforms. A label of "No Hit" means that the sequence should be investigated further using NCBI BLAST[6] to determine whether it is a biologically relevant sequence, or whether it should be removed. The DE also has several versions of BLAST available. The DE BLASTn app is available at: https://de.iplantcollaborative.org/de/?type=apps&amp;app-id=6f94cc92-6d28-45c6-aef1-036be697671d.

After raw sequencing have been screened to produce high-quality reads, the reads need to be assembled to create contiguous sequences (contigs). In brief, assemblies are created by aligning all of the short sequence reads to find similar sequences. Areas of similar sequence larger than a certain length are considered to be the same sequence because the probability of a randomly occurring similar sequence of a certain length is nearly zero. Trinity will output log files, fasta files for each step in the assembly process. However, the most important output is the final assembly file containing the contigs, which is labeled "Trinity.fasta" and found in the main folder. This file contains all of the assembled contigs, and in itself is not practically "human-readable." Therefore, the rnaQUAST tool can be used to understand the assembly in more depth. The rnaQUAST tool will output figures that will allow users to compare assemblies to determine which are most complete (**Figure 4**). Additional information about each figure from rnaQUAST can be found on the wiki (https://wiki.cyverse.org/wiki/x/fwuEAQ). If BUSCO[7] was run, of particular interest is the specificity.txt file which shows the number of complete and partial BUSCO genes and the number of GeneMarkS-T gene predictions in an assembly. BUSCO genes are a curated sets of genes common to a group of organisms. They can be used to assess how well an assembly is capturing sets of genes that are expected to be present in any given type of organism, which is based on phylogenetic clades. A standalone BUSCO app is also available in the DE (https://de.iplantcollaborative.org/de/?type=apps&amp;app-id=112b8a52-efd8-11e5-a15c-277125fcb1b1).

Differential gene expression analysis identifies transcripts that have different patterns of expression across treatments from simple counts per assembled transcript tables. DESeq2 uses a generalized linear model (GLM) to determine variation from a normalized mean. Experiments with replicates are preferred so that technical variation from sequencing can be normalized by the DESeq2 algorithm. DESeq2 DEG analysis yields figures and an .html report file that contains all of the output figures and a description. Alternatively, EdgeR can be used instead of DESeq2, and the same .html report will be generated with EdgeR visualizations instead. Researchers may wish to run both DESeq2 and EdgeR to find differentially expressed genes identified by both algorithms for any given experiment. Trinotate will create an output .xls file that can be opened in any spreadsheet software program. The DEG .txt files and the annotation .xls file can be analyzed and visualized in numerous downstream applications that exist outside the CyVerse platform.

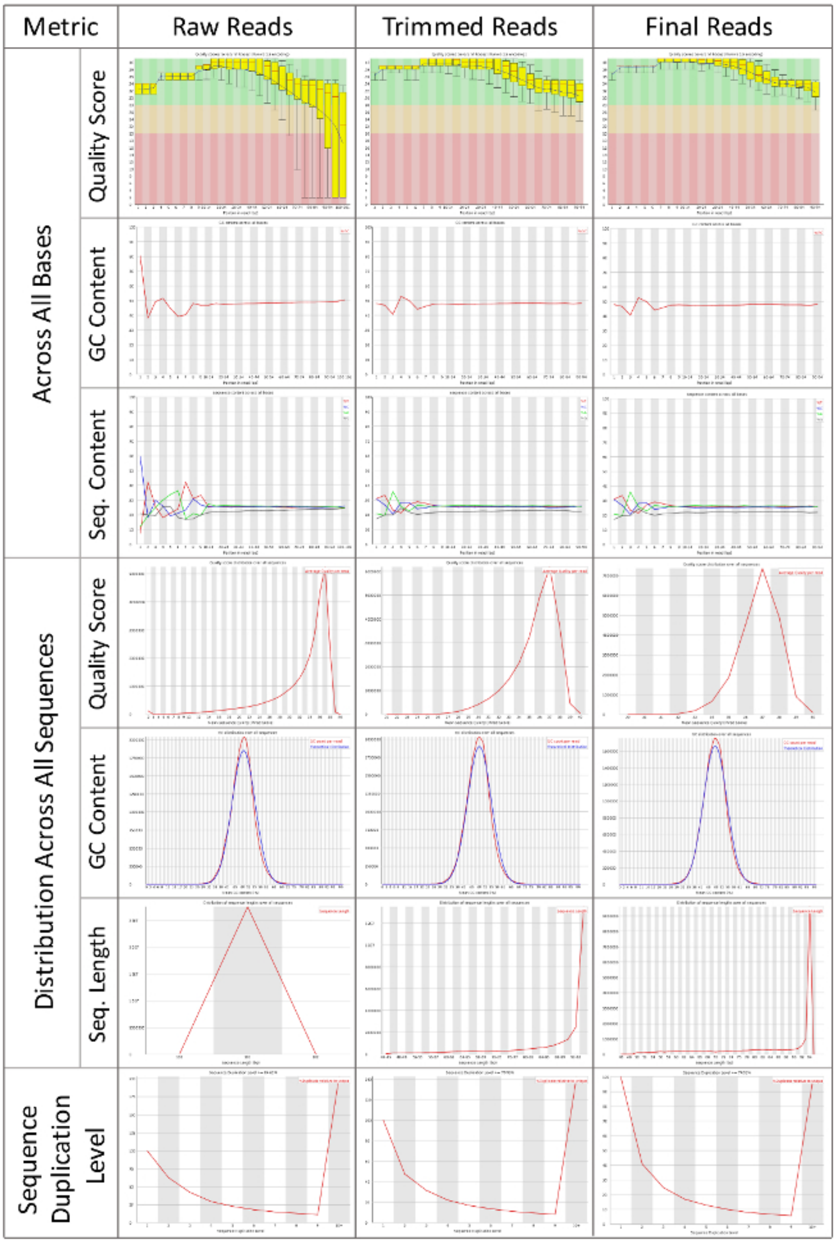

Figure 3: FastQC Reports of Raw Sequencing Reads, Trimmed Reads, and Final Trimmed and Filtered Reads. Systematic comparison of sequencing reads after each pre-processing step. High quality reads are necessary to assemble *de novo* transcriptomes. FastQC can help researchers to understand the initial quality of their sequencing data, and track how efficiently the reads have been pre-processed. Results from FastQC will depend the organisms and samples being sequenced, but uniformity across all samples that will be compared downstream is the primary goal of pre-processing reads. A tutorial video and documentation are available from the authors and developers of FastQC. Please click here to view a larger version of this figure.

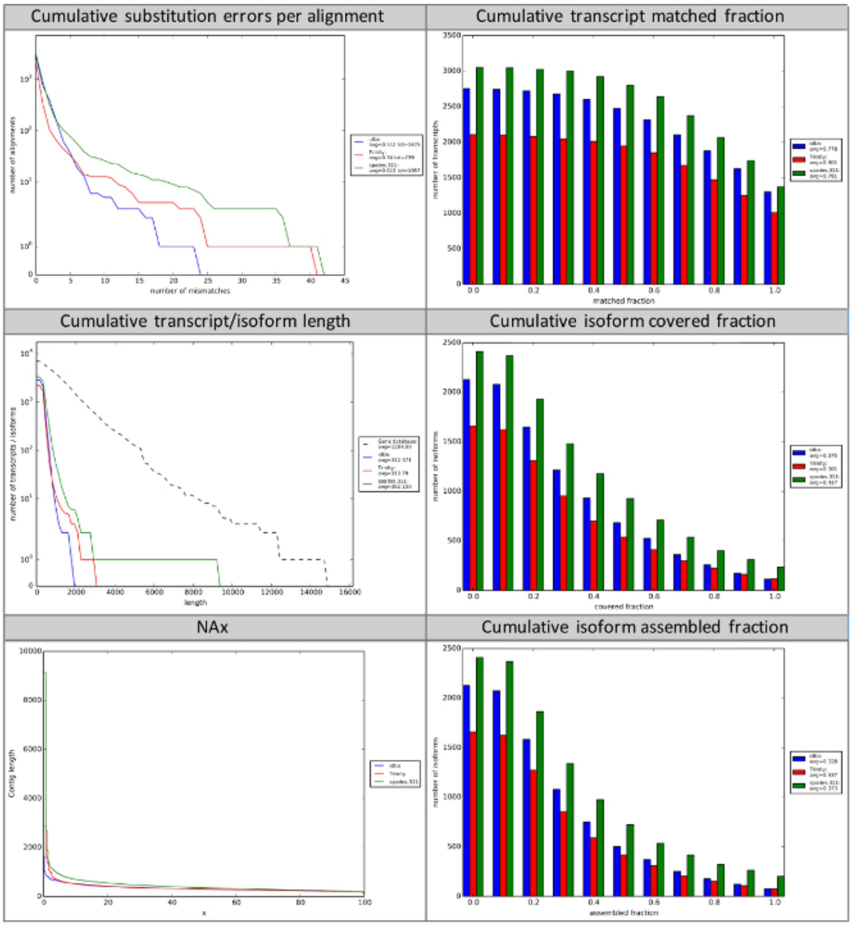

Figure 4: rnaQUAST Reports of Three Separate Assemblies. rnaQUAST can be used to compare multiple read assemblies using the same assembler, or multiple assemblers using the same initial reads. rnaQUAST leverages BUSCO to generate summary statistics about assemblies based on known core genes present in taxonomic clades. The number of mismatches per transcript and how many transcripts match to canonical genes, matched fraction, provide insight into accuracy of assemblers. The last four subplots presented here provide summary statistics of contig and isoform length and the coverage of expected isoforms. NAx represents the percentage (x) of contigs with a length longer than the length (bp) on the y-axis. Assembled fraction is the longest single assembled transcript divided by its length. Covered fraction is the percentage of complete assembled transcripts/isoforms as expected by the core prokaryotic or eukaryotic genes from BUSCO. A description of all graphs generated by rnaQUAST is available (https://wiki.cyverse.org/wiki/x/fwuEAQ). Please click here to view a larger version of this figure.

App Name	CyVerse Platform	Third-party Documentation	CyVerse Documentation	Estimated Runtime for Sample Data Set	Link to App
FastQC	DE	http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ https://www.youtube.com/watch?v=bz93ReOv87Y	https://wiki.cyverse.org/wiki/pages/viewpage.action?pageId=9316768	15 min	https://de.iplantcollaborative.org/de/?type=apps&app-id=112b9aa8-c4a7-11e5-8209-5f3310948295
Trimmomatic v0.33	DE	https://github.com/timflutre/trimmomatic	https://wiki.cyverse.org/wiki/display/DEapps/Trimmomatic-programmable-0.33	30 min	https://de.iplantcollaborative.org/de/?type=apps&app-id=9c2a30dc-028d-11e6-a915-ab4311791e69
Sickle	DE	https://github.com/najoshi/sickle	https://wiki.cyverse.org/wiki/display/DEapps/Sickle-quality-based-trimming	30 min	https://de.iplantcollaborative.org/de/?type=apps&app-id=68b278f8-d4d6-414d-9a64-b685a7714f7c
Trinity	Atmosphere	https://github.com/trinityrnaseq/trinityrnaseq/wiki	https://pods.iplantcollaborative org/wiki/display/atmman/Trinity+-+Trinotate+Atmosphere+Image	1 week	https://atmo.iplantcollaborative.org/application/images/1261
	DE		https://wiki.cyverse.org/wiki/display/DEapps/Trinity-64GB-2.1.1	2-5 days	https://wiki.cyverse.org/wiki/display/DEapps/Trinity-64GB-2.1.1
rnaQUAST v1.2.0	DE, Atmosphere	http://spades.bioinf.spbau.ru/rnaquast/release1.2.0/manual.html	https://pods.iplantcollaborative org/wiki/display/TUT/rnaQUAST+1.2.0+%28denovo+based%29+using+DE	30 min	https://de.iplantcollaborative.org/de/?type=apps&app-id=980dd11a-1666-11e6-9122-930ba8f23352
Transdecoder	DE	https://transdecoder.github.io	https://wiki.cyverse.org/wiki/display/DEapps/Transcript+decoder+2.0	2-3 hours	https://de.iplantcollaborative.org/de/?type=apps&app-id=5a0ba87e-b0fa-4994-92a2-0d48ee881179
DESeq2	DE	https://bioconductor.org/packages/release/bioc/html/DESeq2.html	https://pods.iplantcollaborative org/wiki/pages/viewpage.action?pageId=28115142	2-3 hours	https://de.iplantcollaborative.org/de/?type=apps&app-id=9574e87c-4f90-11e6-a594-008cfa5ae621
EdgeR	DE	https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeR.pdf	https://wiki.cyverse.org/wiki/pages/viewpage.action?pageId=28115144	2-3 hours	https://de.iplantcollaborative.org/de/?type=apps&app-id=4a08ceda-54fe-11e6-862f-008cfa5ae621
Trinotate	Atmosphere	https://trinotate.github.io/	https://pods.iplantcollaborative	1 week	https://atmo.iplantcollaborative.

			org/wiki/display/ atmman/Trinity +-+Trinotate +Atmosphere+Image		org/application/ images/1261

Table 1: Analysis Programs, Platforms they are Available on, and Additional Resources Available for the Workflows in Order by First Appearance. All Package versions are current as of April 2016.

Discussion

There are five critical steps in the protocol that will each create their own separate folder inside of the main project folder (**Figures 1** and **2**). All of the primary raw sequencing data is sacrosanct: it should be uploaded and kept in the first folder labeled "1_Raw_Sequence" and not altered in any way. Data can be uploaded in one of three ways. The DE interface can be used to upload files directly. This is the easiest way to upload data, but also will take the longest to transfer. Cyberduck has a graphical interface and allows users to drag and drop files to transfer to the DE. iCommands is a command line tool that can be used to transfer data to and from the Data Store, make directories and manage data sets, and is likely the fastest way to transfer data files. All data in the Data Store can be shared with other CyVerse users (https://wiki.cyverse.org/wiki/display/DEmanual/Sharing+Data+Files+and+Folders+Via+the+Discovery+Environment), made public through a generated URL (https://wiki.cyverse.org/wiki/display/DEmanual/Sharing+Data+Files+Via+Public+Links), or can be hosted as publicly and anonymously (no username required) available community data (http://data.iplantcollaborative.org; http://mirrors.cyverse.org). Inside that folder, the raw sequence reads are analyzed with FastQC (http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/) to assess how to trim and filter the reads to generate high quality reads. After trimming and quality filtering it is useful to compare the FastQC outputs to determine if the read quality has changed to determine that it has gotten better without losing information (**Figure 3**). Note that the x-axis of FastQC is not linear, but rather is binned for many output graphs, which may lead to misinterpretation of results. The trimmed and filtered reads are then used to assemble *de novo* transcriptomes using an Atmosphere cloud computing instance. This cloud computer uses the local computer screen, keyboard, and mouse, but has its own software (Trinity and Trinotate) and hardware installed. Running programs on the cloud computer instance will not affect the local computer in any way. *De novo* assembly and downstream annotation will most likely be the two longest running steps in this workflow. Therefore, they are completed on Atmosphere to avoid common lab-shared computer problems that would interrupt the analysis like power outages, restarts after late night automatic updates, or crashes caused by other users. Trinotate annotation uses BLAST+[8], HMMER[9], tmHMM[10], and PFAM[11]. The final output of annotation is a SQLite database and a .xls file. The outputs can be used outside CyVerse in downstream analysis platforms such as KEGG[12,13].

This workflow is ready to use in the DE and Atmosphere. This eliminates the need to spend time installing, configuring, and troubleshooting each analysis package and all the dependencies each tool requires. This streamlines researchers' analyses, minimizes wasted effort, and lowers the barrier of entry for many scientists. This workflow specifically assembles either single- or paired-end reads from the Illumina sequencing platform, but many tools exist in the DE and Atmosphere to handle other kinds of sequencing technologies. Tools in this workflow can be easily replaced with a corresponding alternate tool to handle any type of incoming sequencing technology. This is also true of new versions of analysis tools or completely new tools.

This workflow is specifically designed to assemble, compare, and annotate only a few transcriptomes at a time. Therefore, users may find it time consuming to assemble multiple transcriptomes for comparative population genetics. Analysis pipelines will be available to population genetics users in the near future and the link to the pipeline can be found on the wiki page (https://wiki.cyverse.org/wiki/x/dgGtAQ). The differential gene expression analysis step can handle replicates, but it is a pairwise comparison and will not accurately assess multiple factors (*e.g.*, conditions that vary over time, more than two treatments). Automated workflows exist for organisms with reference genomes (*e.g.*, TRAPLINE[14]). While automated workflows are the easiest to use for novices, *de novo* assemblies require assessment and consideration for each step outlined here. Additionally, users are required to use automated pipelines as they are constructed, and therefore are inherently not flexible to meet the changing demands of users.

As most of this protocol is carried out over the Internet, users may experience troubles with their browser settings. Firstly, pop-up blockers may keep windows from opening at all, or may keep windows from opening until permission is given to CyVerse in the browser. Atmosphere uses VNC for accessing remote desktops, but other software may be used. This entire protocol was conducted in Firefox version 45.0.2 and should work with all popular Internet browsers, but some inconsistencies may appear. The workflow will be updated as Trinity releases new versions (https://github.com/trinityrnaseq/trinityrnaseq/wiki). The newest versions and up-to-date information about the workflow can be found on the wiki tutorial page (**Table 1**, https://wiki.cyverse.org/wiki/x/dgGtAQ). Users can contact support directly or post questions at Ask CyVerse (ask.cyverse.org/) to troubleshoot any problems with the workflow.

In the DE several apps exist to accomplish each step of this protocol. For example, users may wish to run Scythe (https://github.com/najoshi/sickle) instead of Trimmomatic[15] for read trimming or run EdgeR[16] instead of DESeq[17,18]. Though outside of the scope of this manuscript, DE apps can be copied, edited, and released by users (https://wiki.cyverse.org/wiki/display/DEmanual/Creating,+Copying,+and+Editing+DE+Apps) or new apps can be added by users (https://wiki.cyverse.org/wiki/display/DEmanual/Dockerizing+Your+Tools+for+the+CyVerse+Discovery+Environment). The Atmosphere images can also be modified and reimaged to create new or modified workflows that match users' needs more specifically (https://wiki.cyverse.org/wiki/x/TwHX). This work serves as an introduction to utilizing the command line to move data and execute analyses. Users can consider utilizing more advanced command line resources such as CyVerse application programming interfaces (APIs) (http://www.cyverse.org/science-apis), or designing their own DE apps, which require knowledge about how the analysis tool is run on the command line (https://wiki.cyverse.org/wiki/display/DEmanual/Creating+a+New+App+Interface).

Disclosures

The authors have nothing to disclose.

Acknowledgements

The authors would like to acknowledge funding from USDA-NIFA grant 2013-00984, NSF grant IOS - 1339156, IOS - 1444490, and CyVerse (NSF: DBI - 1265383).

References

1. Hasselmann, M., Ferretti, L., & Zayed, A. Beyond fruit-flies: population genomic advances in non-Drosophila arthropods. *Brief. Funct. Genomics.* **14** (6), 424-431 (2015).
2. Scholz, M. B., Lo, C.-C., & Chain, P. S. Next generation sequencing and bioinformatic bottlenecks: the current state of metagenomic data analysis. *Anal. Biotech.* **23** (1), 9-15 (2012).
3. Merchant, N., *et al.* The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences. *PLoS Biol.* **14** (1), e1002342 (2016).
4. Oliver, S. L., Lenards, A. J., Barthelson, R. A., Merchant, N., & McKay, S. J. Using the iPlant collaborative discovery environment. *Cur. Protoc. Bioinformatics.* 1-22 (2013).
5. Skidmore, E., Kim, S., Kuchimanchi, S., Singaram, S., Merchant, N., & Stanzione, D. iPlant atmosphere: a gateway to cloud infrastructure for the plant sciences. *Proc. 2011 ACM.* 59-64 (2011).
6. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. Basic local alignment search tool. *J. Mol. Bio.* **215** (3), 403-410 (1990).
7. Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., & Zdobnov, E. M. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics.* (2015).
8. Camacho, C., *et al.* BLAST+: architecture and applications. *BMC Bioinformatics.* **10**, 421-421 (2009).
9. Eddy, S. R. Profile hidden Markov models. *Bioinformatics.* **14** (9), 755-763 (1998).
10. Krogh, A., Larsson, B., von Heijne, G., & Sonnhammer, E. L. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *J. Mol. Biol.* **305** (3), 567-580 (2001).
11. Finn, R. D., Coggill, P., *et al.* The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.* **44** (D1), D279-D285 (2016).
12. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., & Tanabe, M. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.* **44** (D1), D457-D462 (2016).
13. Kanehisa, M., & Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28** (1), 27-30 (2000).
14. Wolfien, M., *et al.* TRAPLINE: a standardized and automated pipeline for RNA sequencing data analysis, evaluation and annotation. *BMC Bioinformatics.* **17**, 21 (2016).
15. Bolger, A. M., Lohse, M., & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics.* **30** (15), 2114-2120 (2014).
16. Robinson, M. D., McCarthy, D. J., & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics.* **26** (1), 139-140 (2010).
17. Anders, S. Analysing RNA-Seq data with the DESeq package. *Mol. Biol.* **43** (4), 1-17 (2010).
18. Love, M. I., Huber, W., & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Bio.* **15** (12), 1-21 (2014).