Improved RNA-seq Workflows Using CyVerse Cyberinfrastructure

Kapeel M. Chougule,[1,5] Liya Wang,[1] Joshua C. Stein,[1] Xiaofei Wang,[1] Upendra Kumar Devisetty,[3] Robert R. Klein,[4] and Doreen Ware[1,2]

[1]Cold Spring Harbor Laboratory, Cold Spring Harbor, New York
[2]United States Department of Agriculture-Agriculture Research Service, Robert W. Holley Center for Agriculture and Health, Ithaca, New York
[3]CyVerse, BIO5, University of Arizona, Tucson, Arizona
[4]United States Department of Agriculture-Agriculture Research Service, Southern Plains Agricultural Research Center, College Station, Texas
[5]Corresponding author: *kchougul@cshl.edu*

RNA-seq is a vital method for understanding gene structure and expression patterns. Typical RNA-seq analysis protocols use sequencing reads of length 50 to 150 nucleotides for alignment to the reference genome and assembly of transcripts. The resultant transcripts are quantified and used for differential expression and visualization. Existing tools and protocols for RNA-seq are vast and diverse; given their differences in performance, it is critical to select an analysis protocol that is scalable, accurate, and easy to use. Tuxedo, a popular alignment-based protocol for RNA-seq analysis, has been updated with HISAT2, StringTie, StringTie-merge, and Ballgown, and the updated protocol outperforms its predecessor. Similarly, new pseudo-alignment-based protocols like Kallisto and Sleuth reduce runtime and improve performance. However, these tools are challenging for researchers lacking command-line experience. Here, we describe two new RNA-seq analysis protocols, in which all tools are deployed on CyVerse Cyberinfrastructure with user-friendly graphical user interfaces, and validate their performance using plant RNA-seq data.

Keywords: bioinformatics • CyVerse • differential gene expression • discovery environment • pseudo-alignment based transcript quantification • reference guided gene expression • RNA-seq • transcript assembly • tuxedo protocol

How to cite this article:
Chougule, K. M., Wang, L., Stein, J. C., Wang, X., Devisetty, U. K., Klein, R. R., & Ware, D. (2018). Improved RNA-seq workflows using cyverse cyberinfrastructure. *Current Protocols in Bioinformatics*, *63*, e53. doi: 10.1002/cpbi.53

INTRODUCTION

The exponential growth and continuously decreasing cost of next-generation sequencing (NGS) technologies have resulted in the creation of large-scale sequencing datasets, especially from RNA sequencing (RNA-seq) studies. In an RNA-seq experiment, mRNA is isolated from a sample of interest and converted to cDNA, which serves as input to an NGS library preparation method. The ease and standardization of protocols for RNA extraction from sample tissue, along with efficient methods for library construction, have made RNA-seq a workhorse for transcriptomic studies. Increasingly, this approach is feasible even for small labs with limited resources.

The resultant deluge of data has increased the demand for user-friendly, reproducible, and scalable tools for analysis of RNA-seq studies (Yang & Kim, 2015). As DNA

CURRENT PROTOCOLS
A Wiley Brand
Current Protocols in Bioinformatics e53, Volume 63
Published in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpbi.53

sequencing technologies have evolved over the years, from Sanger to short read–based Illumina, to single-molecule PacBio long-reads sequencing, the tools for analysis of these data have also evolved with more tools developed for scaling on minimal-to-robust cyberinfrastructure (Niedringhaus, Milanova, Kerby, Snyder, & Barron, 2011).

Current ecosystems of RNA-seq tools provide diverse workflows for analyzing RNA-seq data. Depending on the experimental goal, reads can be aligned to the reference genome or pseudoaligned to the transcriptome, followed by quantification and determination of differential gene expression. Alternatively, if an objective is to annotate the reference, then RNA-seq reads can be assembled using a *de novo* transcriptome assembler (Conesa et al., 2016). The most commonly cited and widely used workflow is the Tuxedo protocol (Trapnell et al., 2012). The main drawback of this workflow is its limited ability to scale (i.e., long runtimes); however, the newly updated Tuxedo protocol has made a marked improvement in its ability to scale (Pertea, Kim, Pertea, Leek, & Salzberg, 2016). The updated Tuxedo protocol not only scales, but also more accurately detects differentially expressed genes. In the case of alignment-free quantification methods, the time-consuming alignment step is skipped; it is replaced by pseudo-alignment in the case of the Kallisto method (Bray, Pimentel, Melsted, & Pachter, 2016), in which a read is assigned to a target sequence without any base-level sequence alignment. Thus, alignment-free methods work much faster than alignment-based methods, but the former cannot be used to identify novel transcripts in an experiment. Most of these tools have been benchmarked with human RNA-seq data or simulated datasets (Sahraeian et al., 2017). In this unit, we will benchmark with actual plant RNA-seq data from a study that compared the drought response of two sorghum genotypes with different water use efficiencies (Fracasso, Trindade, & Amaducci, 2016).

In this unit, we will examine two newly integrated RNA-seq workflows in the CyVerse cyberinfrastructure (Goff et al., 2011) (Fig. 1). Basic Protocol 1 describes the following: the new Tuxedo protocol, which uses HISAT2 as a splice-aware aligner to align short reads to the reference genome (Kim, Langmead, & Salzberg, 2015); StringTie (Pertea et al., 2015) to assemble transcripts from read-aligned BAM (Li et al., 2009) files; StringTie-merge to combine assembled transcripts into a consolidated annotation set; and the Ballgown R package to determine differential expression (Frazee et al., 2015). All apps are available in an easy-to-use GUI interface on the CyVerse Discovery Environment platform. Basic Protocol 1 represents an improvement over older Tuxedo framework that used TopHat, Cufflinks, Cufflinks-merge, and Cuffdiff. Basic Protocol 2 describes an alternative RNA-seq analysis protocol, which relies on pseudo-alignment for quantification of transcripts, using Kallisto, and the Sleuth R package (Pimentel, Bray, Puente, Melsted, & Pachter, 2017) for differential expression analysis. Both apps again are available in an easy-to-use GUI interface on the CyVerse Discovery Environment platform.

For both protocols, we will use *Sorghum bicolor* RNA-seq data (Fracasso et al., 2016) comparing transcript abundance in the drought sensitive genotype IS20351 under drought stress (DS) and well-watered (WW) conditions. Using the RNA-seq data, we will compare gene expression levels between the DS and WW conditions and identify new transcripts or isoforms. The tutorial will use data stored at the NCBI Sequence Read Archive as described in Table 1.

BASIC PROTOCOL 1 ALIGNMENT-BASED TUXEDO PROTOCOL USING THE CyVerse DISCOVERY ENVIRONMENT: HISAT2, StringTie, StringTie-merge, AND BALLGOWN

Depending on the goal of the experiment, RNA-seq data can be used for quantification of gene or transcript expression, construction of transcripts to improve annotations,

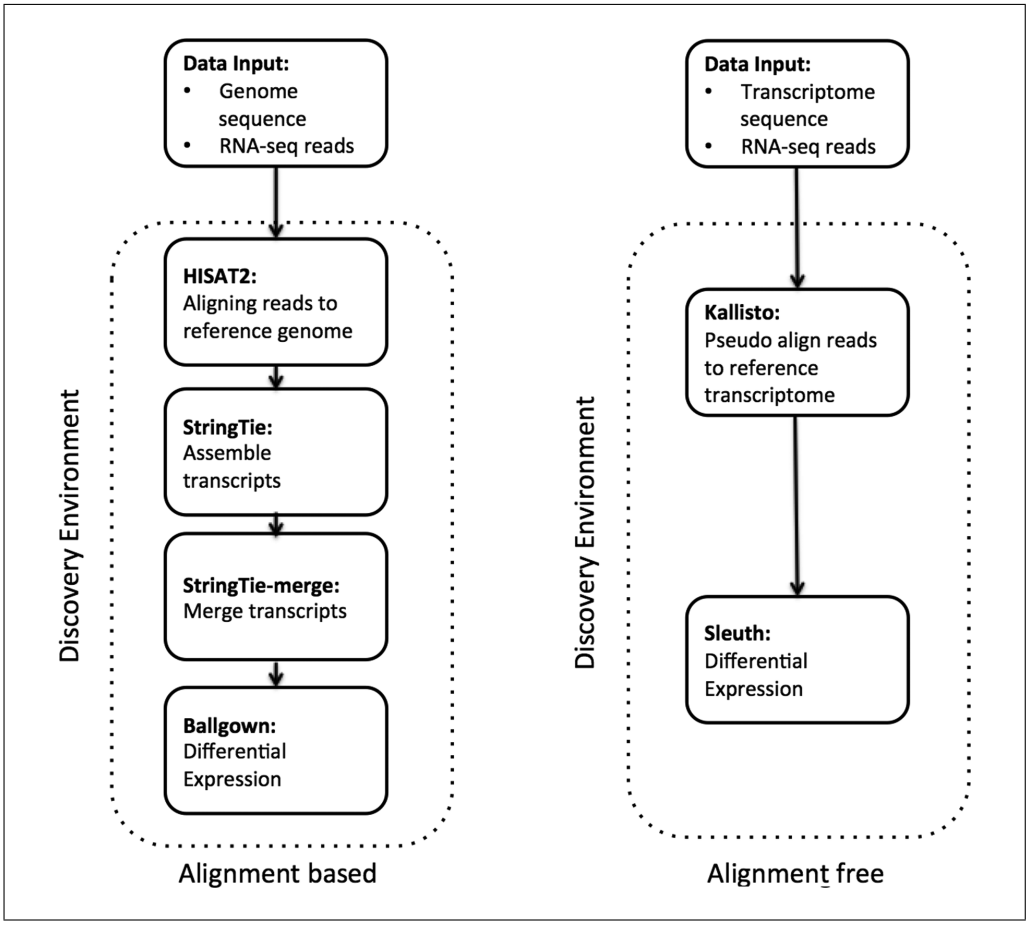

Figure 1 RNA-seq workflows using CyVerse cyberinfrastructure. The workflow includes two protocols: (1) Alignment-based Tuxedo protocol using the CyVerse Discovery Environment: HISAT2, StringTie, StringTie-merge, and Ballgown; (2) Pseudo-alignment based protocol using Kallisto and Sleuth. Detailed procedures are described in the protocols.

Table 1 *Sorghum Bicolor* RNA-seq Data (Fracasso et al., 2016) Comparing Transcript Abundance in the Drought Sensitive Genotype IS20351 Under Drought Stress (DS) and Well-Watered (WW) Conditions

Gene Expression Omnibus ID	Sample name	Link
GSM2133750	IS20351_WW_1	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133750*
GSM2133751	IS20351_WW_2	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133751*
GSM2133752	IS20351_WW_3	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133752*
GSM2133753	IS20351_DS_1	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133753*
GSM2133754	IS20351_DS_2	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133754*
GSM2133755	IS20351_DS_3	*https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2133755*

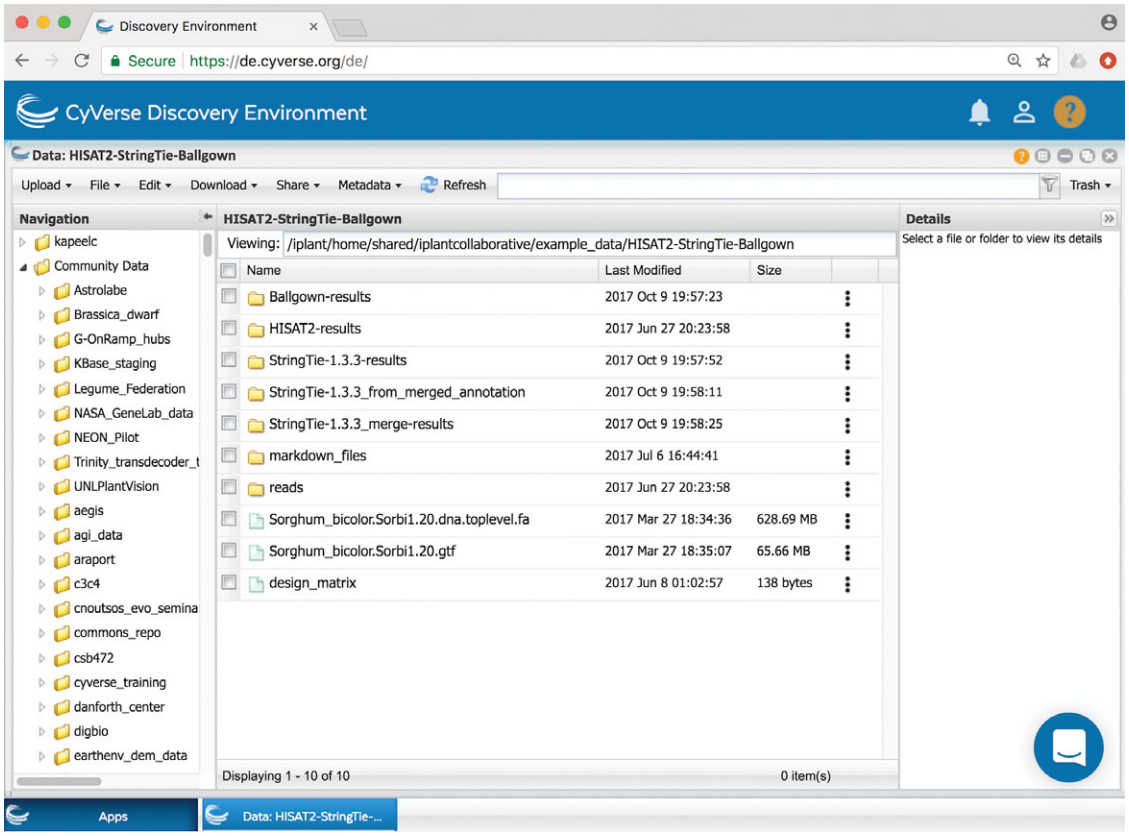

Figure 2 Staged input and output files for Basic Protocol 1.

identification of novel transcripts, or differential quantification of gene or transcript expression based on experimental conditions. Some protocols only align reads to either the reference genome or the transcriptome, and then quantify based on either genes or transcripts. The protocol below does both, and is thus capable of identifying both novel genes and transcript isoforms. The CyVerse Discovery Environment (DE) (Oliver, Lenards, Barthelson, Merchant, & McKay, 2013) offers users a powerful computational platform with access to hundreds of bioinformatic apps with an easy-to-use Web interface. DE categorizes applications based on domain of science, functionality of the apps, and as High Performance Computing apps for large-scale analysis. All apps expect the HPC run on a HTCondor system where each app runs on a dedicated 4-CPU and 8 GB RAM machine. DE offers instant deployment of the apps for users, with ability to launch multiple apps. In addition to launching apps, DE offers the ability to manage and share data files, workflows, analyses of results, and data visualizations with collaborators or securely with the community at large, and make them accessible within the iRODS data management system. Both the older and newer Tuxedo-based protocols are available on DE. Each tool within the protocol is integrated into the platform using Docker container technology (Devisetty, Kennedy, Sarando, Merchant, & Lyons, 2016). The older Tuxedo protocol comprises TopHat2, Cufflinks2, Cuffmerge2, and Cuffdiff2, whereas the newer protocol replaces these tools with HISAT2, StringTie, StringTie-merge, and Ballgown. Each tool in the newer protocol is faster and more accurate when compared to its counterpart in the older protocol. The use of these apps on CyVerse allows the user to load all sample files at once; in addition, an easy-to-use graphical interface is offered for each app.

Necessary Resources

Hardware

A computer with Internet access

Software

An up-to-date Web browser, such as Internet Explorer (*https://www.microsoft.com/ie/*), Firefox (*https://www.mozilla.com/*), Chrome (*https://www.google.com/chrome*), or Safari (*https://www.apple.com/safari/*). JavaScript must be enabled in the Web browser.

Files

Reference genome in FASTA format

RNA reads in FASTQ format (single-end or paired-end); the read file can also be passed in compressed form in gzip (`.gz`) or bzip2 (`.bz2`) format

Input and output files for Basic Protocol 1 are stored on the CyVerse Data Store, accessible via the Discovery Environment (Fig. 2)

The `Sorghum_bicolor.Sorbi1.20.dna.toplevel.fa` file can be obtained from the Gramene project FTP site (*ftp://ftp.ensemblgenomes.org/pub/plants/release-20/fasta/sorghum_bicolor/dna/Sorghum_bicolor.Sorbi1.20.dna.toplevel.fa.gz)*

Reference annotation in GTF format (`Sorghum_bicolor.Sorbi1.20.gtf`; *ftp://ftp.ensemblgenomes.org/pub/plants/release-20/gtf/sorghum_bicolor/Sorghum_bicolor.Sorbi1.20.gtf.gz*

Both files should be unzipped before uploading to the app

Access Discovery Environment

1. Using your CyVerse credentials, log in to the Discovery Environment of the CyVerse Web site at *https://de.cyverse.org* (Fig. 3).

   To access the Discovery Environment, you need to first register at https://user.cyverse.org/register.

Align reads to the reference using the HISAT2 aligner

2. Once you are logged in, click on the Apps button on the left side of the window (Fig. 4) to get the full list of apps available on CyVerse.

3. You can either search for `HISAT2-index-align-2.1` or navigate around the list of apps from the left navigation column to find "HISAT2-index-align-2.1" (Fig. 5; *Categories* → *Operation* → *Alignments* → *HISAT2-index-align-2.1*).

4. Click on this app. Another pop-up window will appear where you will provide your inputs (Fig. 6).

5. In the first section of the app, give a name to your session, write a comment about this particular analysis, and set the name and location of the directory where all of the output files will be deposited (Fig. 7).

6. Click on the Input drop-down menu; there will be a section where you can choose your input files. The first file is the reference genome in FASTA format; the user uploads clicking on the Browse button. The second file contains the reads in FASTQ format. Both single-end and paired-end sequencing reads can be used for the analysis, but the format must be specified. As an option, Fragment Library Type can be provided to describe the orientation of the sequenced reads (Fig. 8).

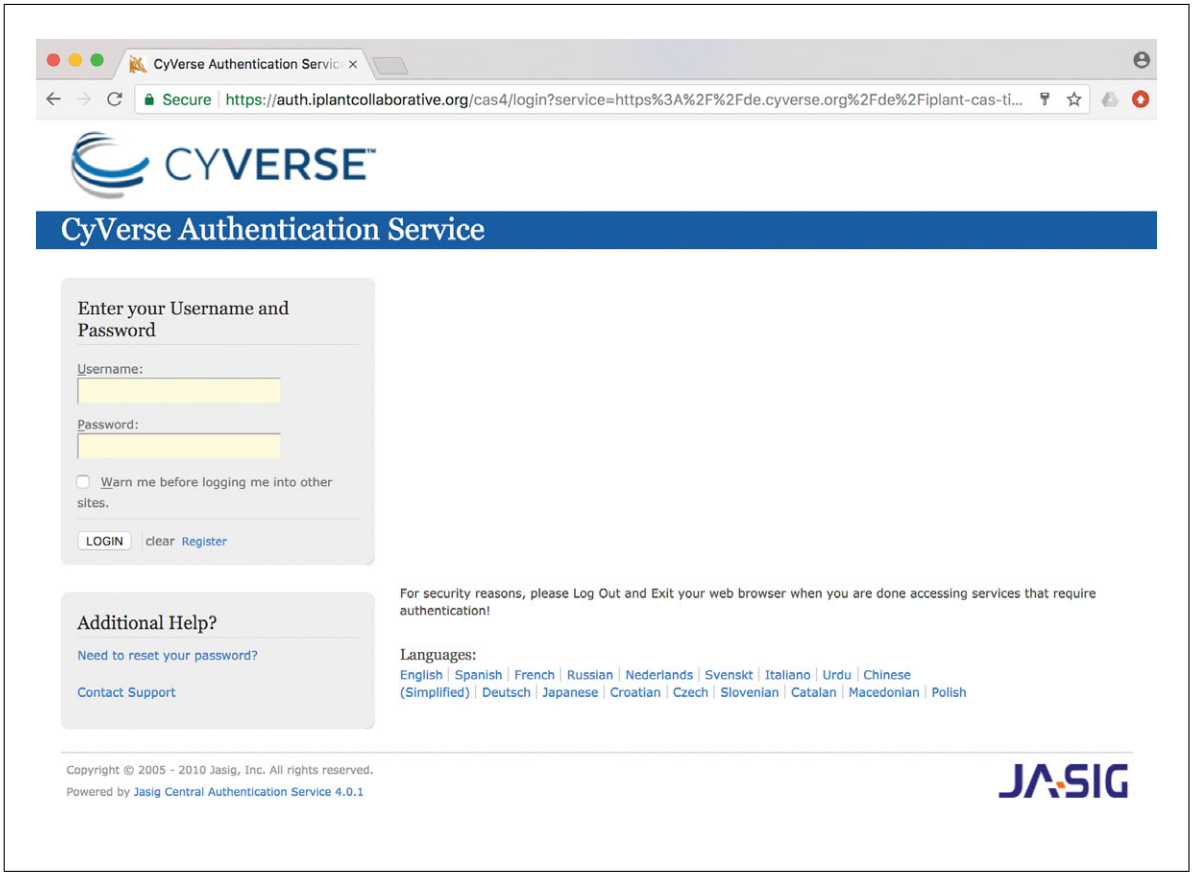

Figure 3 Login screen at CyVerse Discovery Environment.

The app allows the user to upload multiple read files for all the samples. If using paired-end reads, load the `read1` files under option FASTQ Files (Read 1) and `read2` files under FASTQ Files (Read 2). For single-end reads, load all the read1 files under FASTQ Files (Read 1) and leave the FASTQ Files (Read 2) empty.

In our example data, we will use *Sorghum bicolor* v1.2 as the reference genome and paired-end *Sorghum* RNA-seq `fastq.gz` read files as described under Necessary Resources. To upload the reference genome, click on Browse and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown*. Select the file `Sorghum_bicolor.Sorbi1.20.dna.toplevel.fa` and click 'OK'. (Fig. 9). Similarly, load the `read1` and `read2` gzipped FASTQ files for all the samples. Click on Add under FASTQ Files (Read 1), which will open a data window, and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *reads* (Fig. 10). To upload `read1` files click on `read1` folder and select all the `read1` files by checking the box before the file names. The app allows multiple file uploads. The user can select all files at once by checking the box before the Name column (Fig. 10) and click OK. Similarly, upload the `read2` files under FASTQ Files (Read 2); use files in the read2 folder. Make sure that the order of the files in both read1 and read2 inputs is the same.

Note that users can upload their own data using the upload feature described at https://wiki.cyverse.org/wiki/display/DEmanual/Uploading+and+Importing+Data+Items+Within+the+DE, which contains instructions that allow users to bring in their own RNA-seq data in FASTQ format and reference genome in FASTA format and upload them to the data store. Once on the data store, the user can click on the Browse option with the `HISAT2-index-align-2.1` app to provide the input files.

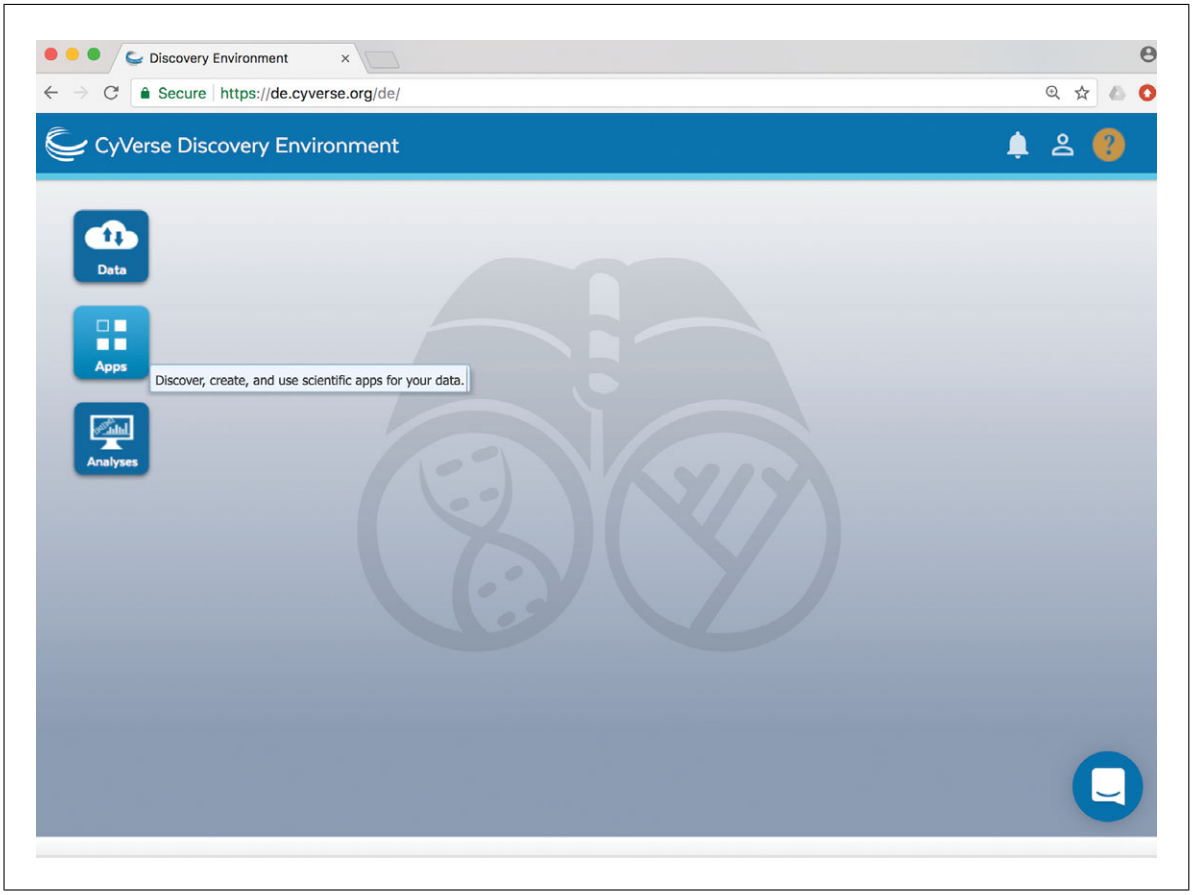

Figure 4 Layout of the Discovery Environment post-authentication step.

The DE Web interface allows the user to upload files of maximum size 1.9 Gb. For uploading files of larger size, and for bulk upload, see Troubleshooting. There is a 100-Gb disk allocation limit for a CyVerse user account; for additional disk space the user can submit a request for data allocation increase at https://user.cyverse.org/forms/2 with the justification for the allocation increase.

7. Leave the Fragment Library Type option blank and select File Type as PE, as our example data reads are unstranded and paired-end. If the user provided reads that are strand specific, select the options describing the orientation of reads from the drop-down list. If the library is single-end, select File Type as SE.

   Fragment Library Type indicates whether the reads are strand-specific or unstranded. This information, usually provided by the sequencing center, indicates which strand of cDNA was used for sequencing of the reads. If the sequencing was strand-specific and paired-end, then the read orientation can be forward-reverse with read 1 being forward and read 2 being reverse. Other modes of orientation include reverse-forward, forward-forward, and (for single-end reads) either forward or reverse. If the sequencing protocol was not strand-specific, the user can leave this option blank; the default value is unstranded

8. Click on the "Advance option" drop-down menu to set all parameters for the run.

   For your convenience, the default values have been set for each field, and clicking on the information button (labeled with an 'i') to the right of an individual parameter will provide further information on that setting. The user can specify a Phred quality score encoding schema for the reads; these are generally of two types, phred33 and phred64. Phred33 is used by the latest Illumina pipelines. Based on the quality score

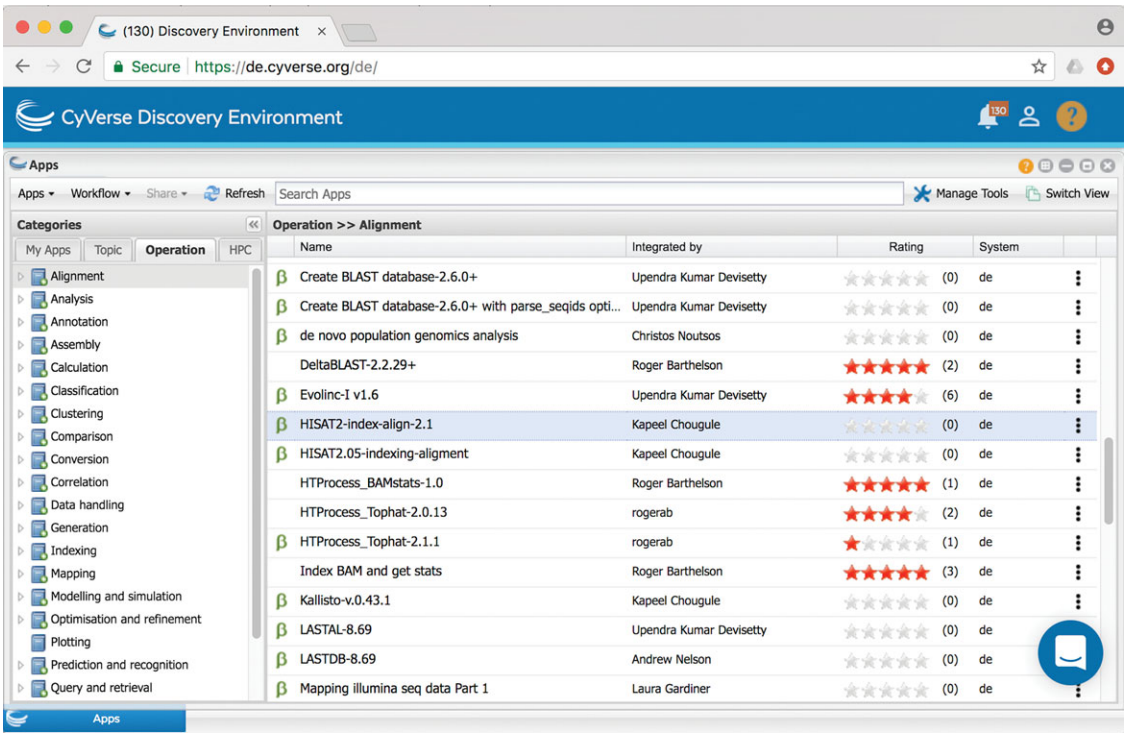

Figure 5 Navigate to find HISAT2-index-align-2.1 on Discovery Environment.

distribution, the user can apply filters to trim low-quality bases from the 5′ and 3′ ends of the reads. It is essential to remove low-quality bases prior to alignment in order to avoid spurious alignment of bases to the reference. The user can set the minimum and maximum values for intron length. These values can be specific to the genome to which the reads are aligned. In previous RNA-seq splicing studies in *Arabidopsis*, these values have been set to 60 and 6000. This can vary across organism; one can get a distribution of intron lengths from the reference annotation. It is necessary to select the option for tailoring the outputs to be used by StringTie for the next step in this protocol. The last two options describe the minimum and maximum fragment lengths, and are kept at the default values (Fig. 11).

9. After making sure that all the input parameter fields have been correctly filled out, click Launch Analysis at the bottom right corner of the app window to run `HISAT2-index-align-2.1`.

   Once the app has launched, you will receive a prompt alerting you that your job has been submitted. As a default, you will receive an e-mail notification when the status of the job changes to 'complete'.

10. To track the status of your analysis, you can click on the Analyses button in the left panel to get the list of analyses you have run on the Discovery Environment and check status by looking at the Status section. Alternatively, you can click on the bell-shaped icon on the top-right panel and see the status of your most recently launched jobs.

11. The time required for the analysis to complete will depend on the size of the genome and sequencing reads, and may be as long as a few hours. Once the job is complete, you will receive an e-mail notification saying that the status of your run has changed from "running" to "completed" (Fig. 12).

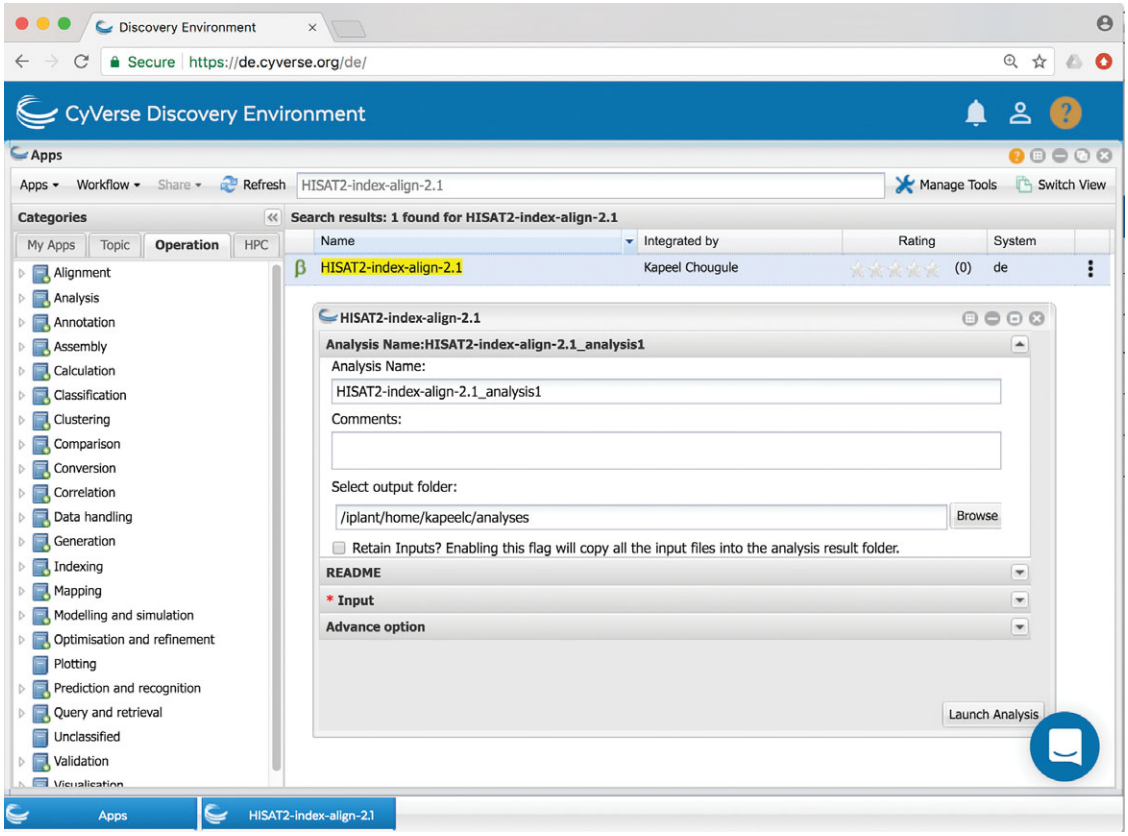

Figure 6 Pop-up window of HISAT2-index-align-2.1 app in the Discovery Environment.

HISAT2 shows a marked improvement in runtime in comparison with its predecessor alignment tool, TopHat2. Like TopHat2, HISAT2 uses one global FM index along with several small local FM indexes to build an efficient data structure with Bowtie2 (Langmead & Salzberg, 2012) as the underlying mapping engine, resulting in an improvement to the alignment speed several times faster than TopHat2.

12. After the job is completed, you can click on the name of the job in the Analyses window and examine the output files. All of the files generated from the analyses are permanently available to the user. The app creates an output directory containing the following output files (Fig. 13):

    - `$PREFIX.sorted.bam`: coordinate-sorted alignment files in `.bam` format for each sample
    - `$PREFIX.sorted.bam.bai`: index files for `.bam` alignments used for visualizing in genome browsers.

    Here `$PREFIX` refers to the base name of the read files. All of these files can be viewed directly from the Discovery Environment or downloaded using the Download drop-down menu in the Data window.

Assemble transcripts with StringTie-1.3.3

StringTie, like Cufflinks2, assembles transcripts from RNA-seq reads aligned to the reference, and also performs quantification. It follows a netflow algorithm in which it simultaneously assembles and quantitates the highly expressed transcripts, removes reads associated with those transcripts, and repeats the process until all reads are used. This algorithm improves the run time of StringTie and uses less memory than Cufflinks2. If

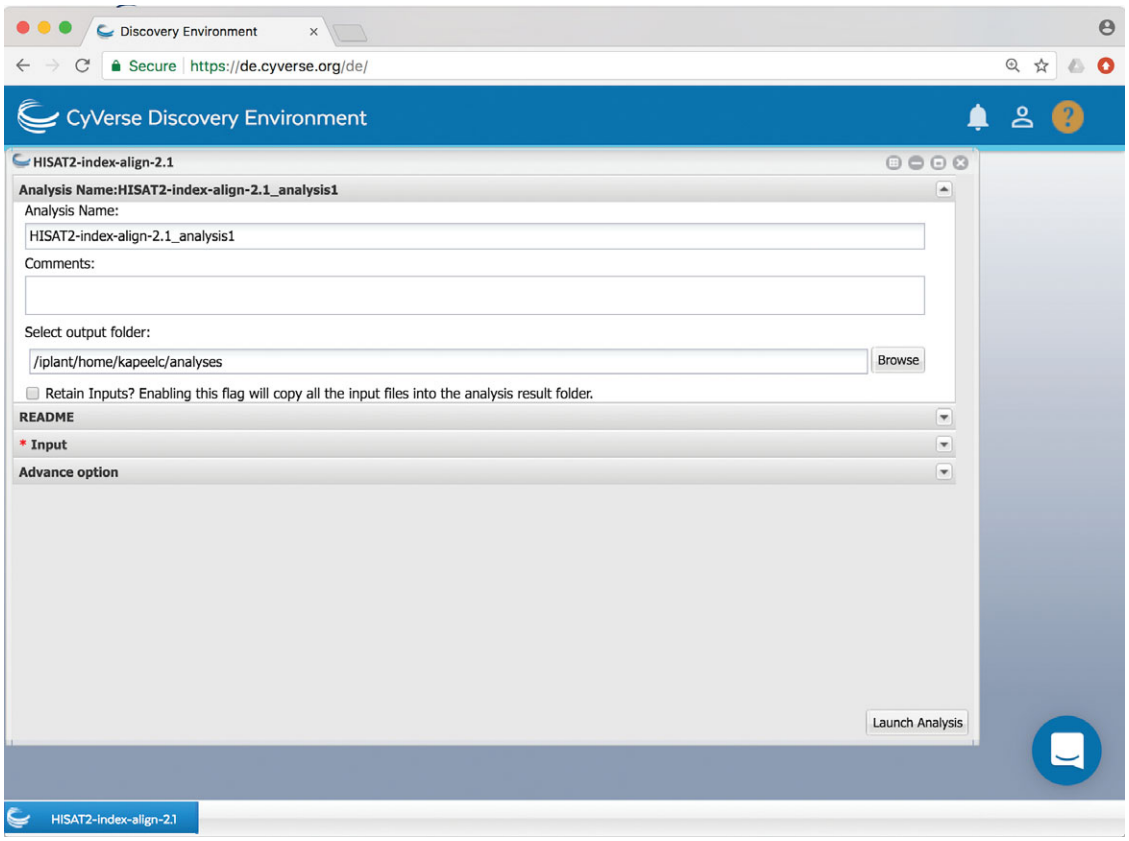

Figure 7 Fields to describe the run, analysis name, comments about the run, and the directory for all output files.

provided with a reference annotation file, StringTie uses it to construct an assembly for low-abundance transcripts, but this is optional. Alternatively, you can skip the assembly of novel genes and transcripts, and instead use StringTie simply to quantify all transcripts provided in an annotation file.

13. Repeat steps 1 to 5, this time searching for StringTie-1.3.3 or navigating to *Categories* → *Operation* → *Assembly* → *StringTie-1.3.3*.

14. StringTie takes as input a binary coordinate-sorted BAM file. This file contains spliced read alignments such as the ones produced by the HISAT2 app. Once you click on the 'Input data' drop-down menu, you will see a section where you can upload your BAM files. Users upload their own BAM file or select from the Discovery Environment by clicking on the Add button. Add the bam files by navigating to the `bam_output` folder from the `HISAT2-index-align-2.1` run described step 12, or by using the pre-staged HISAT2 result bam files under *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *HISAT2-results* → *bam_output*. Select and drag all BAM files into the input box and click OK. For convenience, a batch of BAM files can be analyzed together, but these files can also be processed concurrently in independent StringTie runs (Fig. 14).

15. In the Reference Annotation section, the user provides an annotation file in GTF format or selects from the list of annotations for the relevant species. In our example, we will use the *Sorghum bicolor* v1.2 community annotation, as described under Necessary Resources. Click on Browse, navigate to *Community*

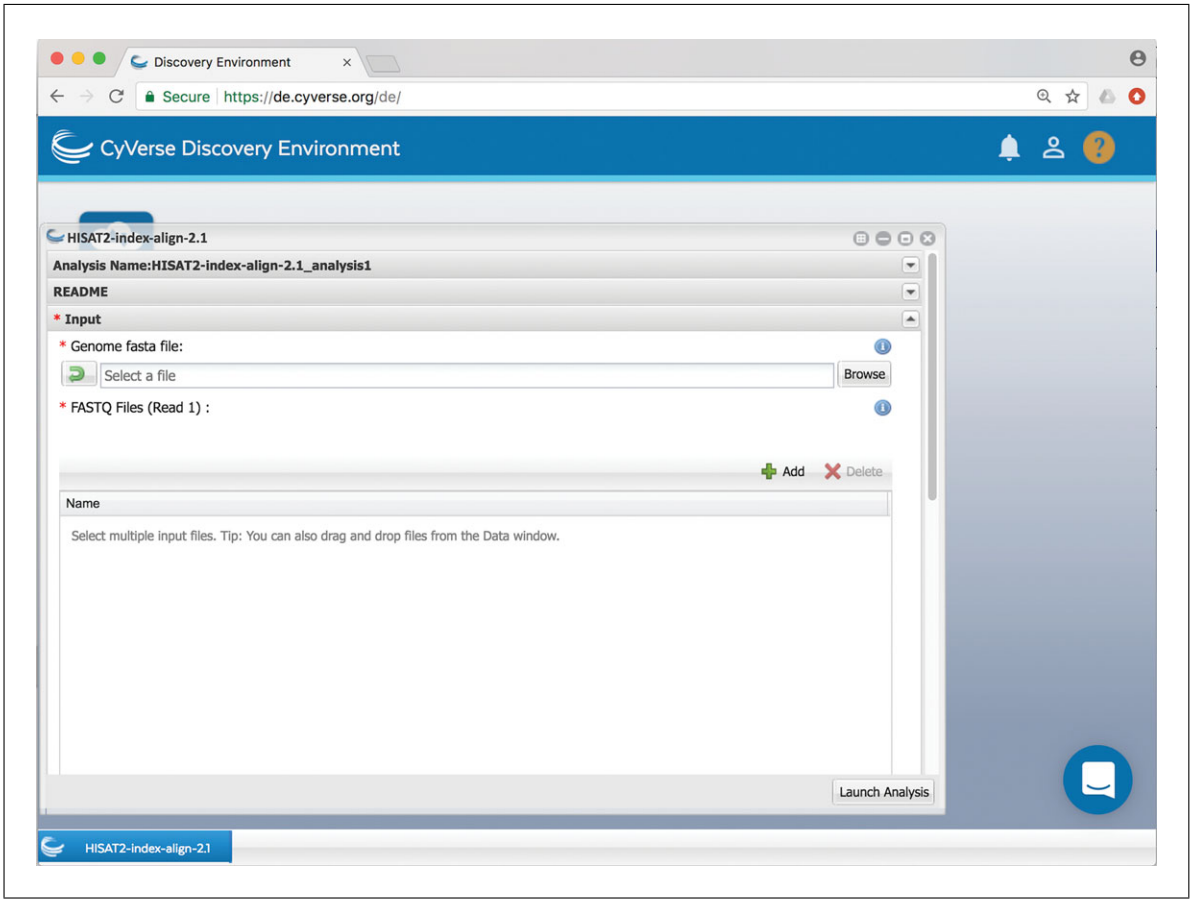

Figure 8 Fields to specify inputs for the HISAT2-index-align-2.1 run. Required input files are the reference genome in `.fasta` or `.fa` format, and either one read file in `.fastq` or compressed format (`.bz2` or `.gz`) for single-end reads or two read files in the same format for paired-end reads.

Data → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *Sorghum_bicolor.Sorbi1.20.gtf*, and click OK (Fig. 15).

This step is optional, but if a high-quality reference annotation is available, it is recommended that you use it in this step. StringTie will check to see whether the reference transcripts are expressed in the RNA-seq data, and for those that are expressed, it will compute coverage and FPKM values. Note that reference transcripts need to be fully covered by reads in order to be included in StringTie's output. Other transcripts assembled from the data by StringTie and not present in the reference file will be printed as well.

16. Under 'Analysis options', users can define their customized parameters for all the options. Default suggestions have been set and clicking on the information button ('i') to the right of an individual parameter will provide further information on that setting (Fig. 16).
17. After making sure that all input parameter fields have been correctly filled out, click Launch Analysis to run StringTie-1.3.3.
18. To track the status of your analysis, you can click on the Analyses button.
19. Depending on the number of samples used for transcript assembly, the StringTie run can take anywhere from 30 min to 1 hr to complete.
20. After the job is completed, you can click on the name of the job from the Analyses window and examine the output files (Fig. 17). The app generates a `StringTie_output` folder containing the following files:

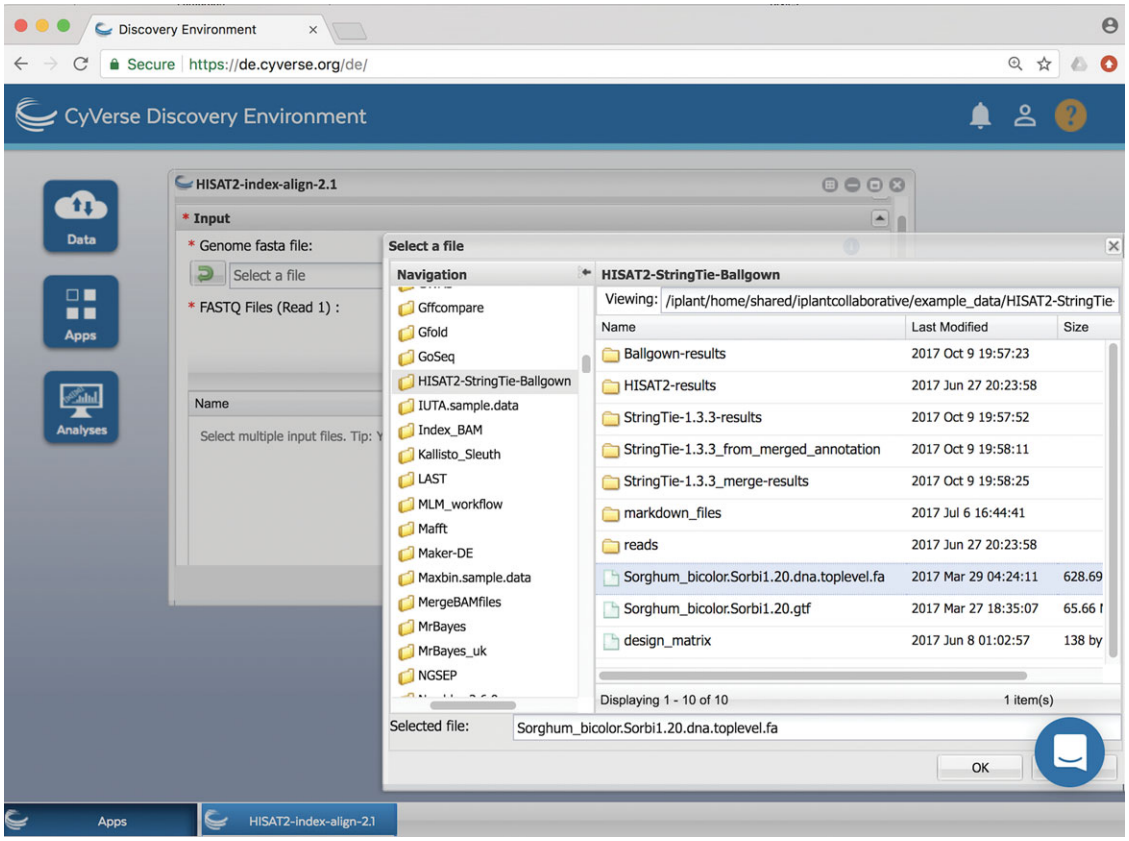

Figure 9 Load reference genome `Sorghum_bicolor.Sorbi1.20.dna.toplevel.fa`.

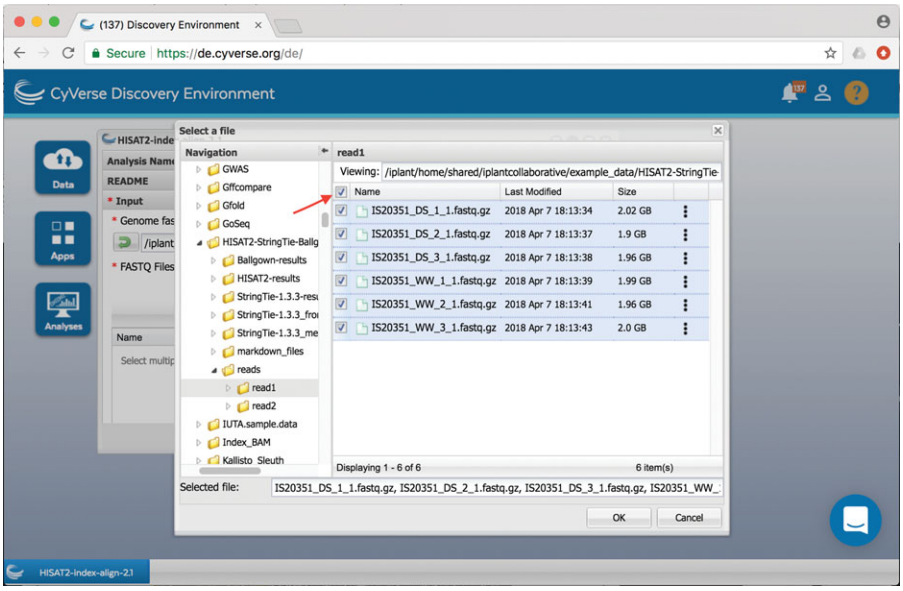

Figure 10 Load multiple paired *Sorghum* RNA-seq read 1 files. Click on the checkbox on left of the Name column shown by red arrow to select all files at once. Similarly upload read 2 files.

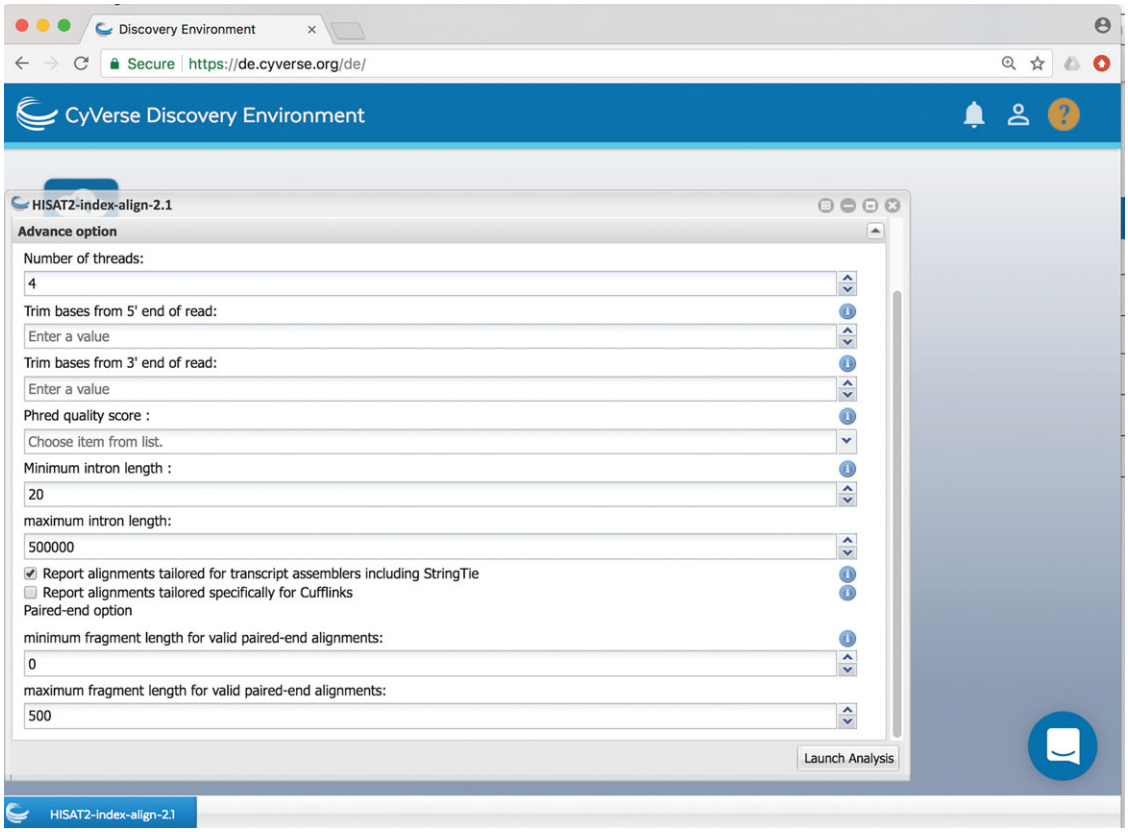

Figure 11 Advanced options for HISAT2-index-align-2.1 app.

- `$PREFIX.gtf`: StringTie's main output is a GTF file containing the assembled transcripts
- `$PREFIX.abund.tab`: Gene abundances in tab-delimited format
- `$PREFIX.refs.gtf`: Fully covered transcripts that match the reference annotation, in GTF format.

> *Here `$PREFIX` refers to the base name of the bam files. Examine the GTF file, `IS20351_DS_1_1.gtf`, which contains annotated transcripts assembled by StringTie-1.3.3. This file gives normalized expression metrics in both FPKM and TPM along with per-base coverage, the values for which can be found in the last column for the file (Fig. 18). A separate folder with just `.gtf` files is to be used for merging transcripts using StringTie-1.3.3-merge. All of these files can be viewed directly from the Discovery Environment or downloaded using the Download drop-down menu in the data window.*

Merge all StringTie-1.3.3 transcripts into a single transcriptome annotation file

StringTie-merge, like Cufflinks-merge, merges transcript assemblies from samples into a consolidated annotation set. This step helps restore complete structures of the transcripts, especially for transcripts assembled with low coverage. The main purpose of this application is to more easily generate an assembly GTF file suitable for use with Ballgown. A merged, empirical annotation file will be more accurate than the standard reference annotation, as the expression of rare or novel genes and alternative splicing isoforms in a given experiment will be better reflected by empirical transcriptome assemblies. The `StringTie-1.3.3_merge` app takes as input a multiple transcript assembly GTF files. These files contain assembly coordinates of transcripts assembled using StringTie-1.3.3. `StringTie-1.3.3_merge` will merge these assembly files into a consolidated annotation file containing a non-redundant set of transcripts. This

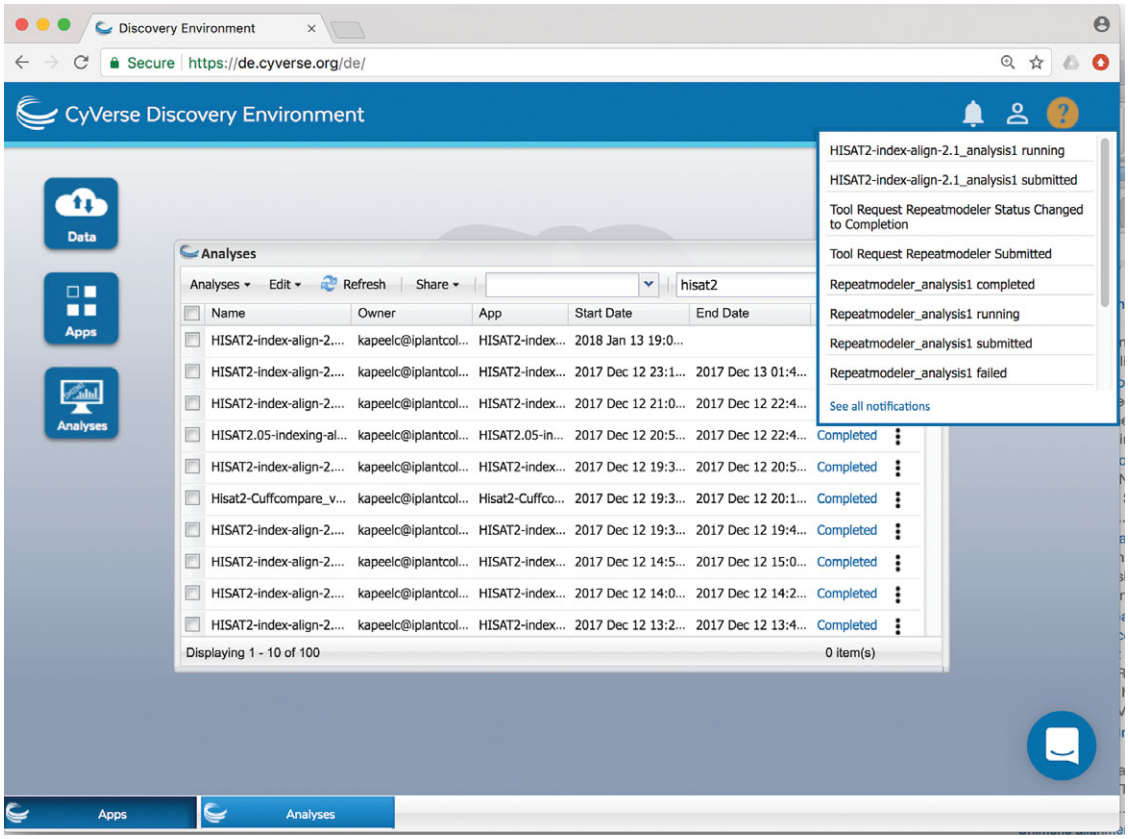

Figure 12 Job status window in the CyVerse Discovery Environment. Once the job runs to completion, it is also possible to check the total run time in this window.

mode is used to generate a global, unified set of transcripts (isoforms) across multiple RNA-seq samples.

21. Repeat steps 1 to 5, in this case searching for `StringTie-1.3.3_merge` or navigating to *Categories* → *Operation* → *Assembly* → *StringTie-1.3.3_merge*.

22. Click on the 'Input data' drop-down menu, you will see a section where you can upload your files. The user uploads their own GTF files or selects from the Discovery Environment by clicking on the Add button. Add the gtf files by navigating to the gtf folder from the StringTie-1.3.3 run described in step 20 or by using the GTF files from the pre-staged `StringTie-1.3.3-results` folder under *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *StringTie-1.3.3-results* → *gtf_files*. Select and drag all gtf files into the input box and click OK (Fig. 19).

23. In the Reference Annotation section, the user provides an annotation file in GTF format or selects from the list of annotations for the relevant species. Click on Browse and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *Sorghum_bicolor.Sorbi1.20.gtf*, and click OK (Fig. 20).

    This step is optional, but if a high-quality reference annotation is available, it is recommended that you use it in this step. `StringTie-1.33_merge` *will assemble the predicted transcripts (transfrags) from the input GTF file containing the reference transcripts.*

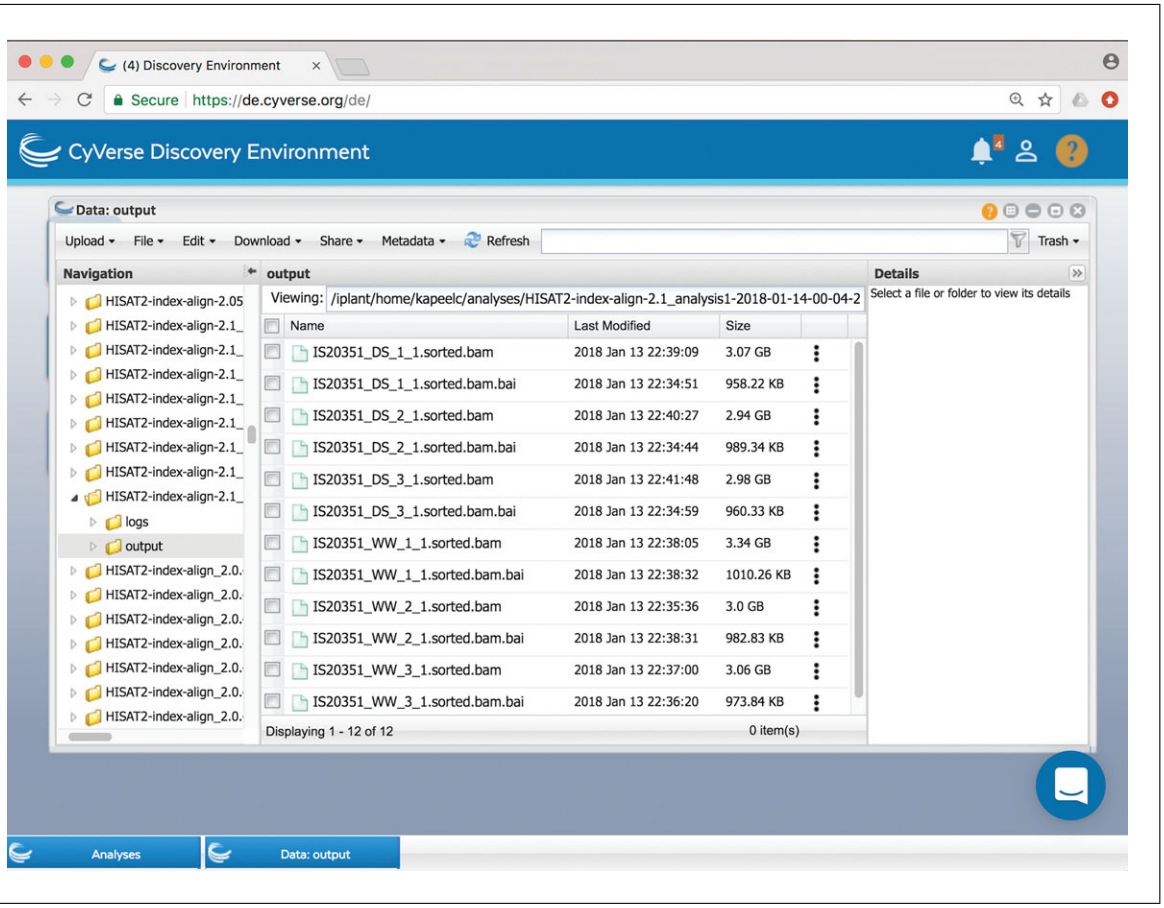

Figure 13 Output folder from `HISAT2-index-align-2.1` run containing `.bam` and `.bai` files for each sample.

24. Under the Analysis option, users can define their customized parameters for all the options; the parameters are currently set to their default values and clicking on the information button ('i') to the right of an individual parameter will provide further information on that setting (Fig. 21).

25. After making sure that all input parameter fields have been correctly filled out, click on 'Launch Analysis' to run `StringTie-1.3.3_merge`.

26. To track the status of your analysis, you can click on the Analyses button.

27. Depending on the number of samples used for transcript assembly, a `StringTie-1.3.3_merge` run can take anywhere from 10 to 15 min to complete.

28. After the job is completed, click on the name of the job in the Analyses window and examine the output files (Fig. 22). The app generates a merged annotation, `merged.out.gtf`, and a consolidated annotation file from transcript assemblies generated by `StringTie-1.3.3_merge`.

    All of these files can be viewed directly from the Discovery Environment or downloaded using the Download drop-down menu in the data window.

Create Ballgown input files using StringTie-1.3.3

Run StringTie-1.3.3 again to assemble transcripts. This time, however, we will use the consolidated annotation file we obtained in the `StringTie-1.3.3_merge` step. This will re-estimate transcript abundances using the merged structures; however, reads may need to be re-allocated for transcripts whose structures were altered by the merging step.

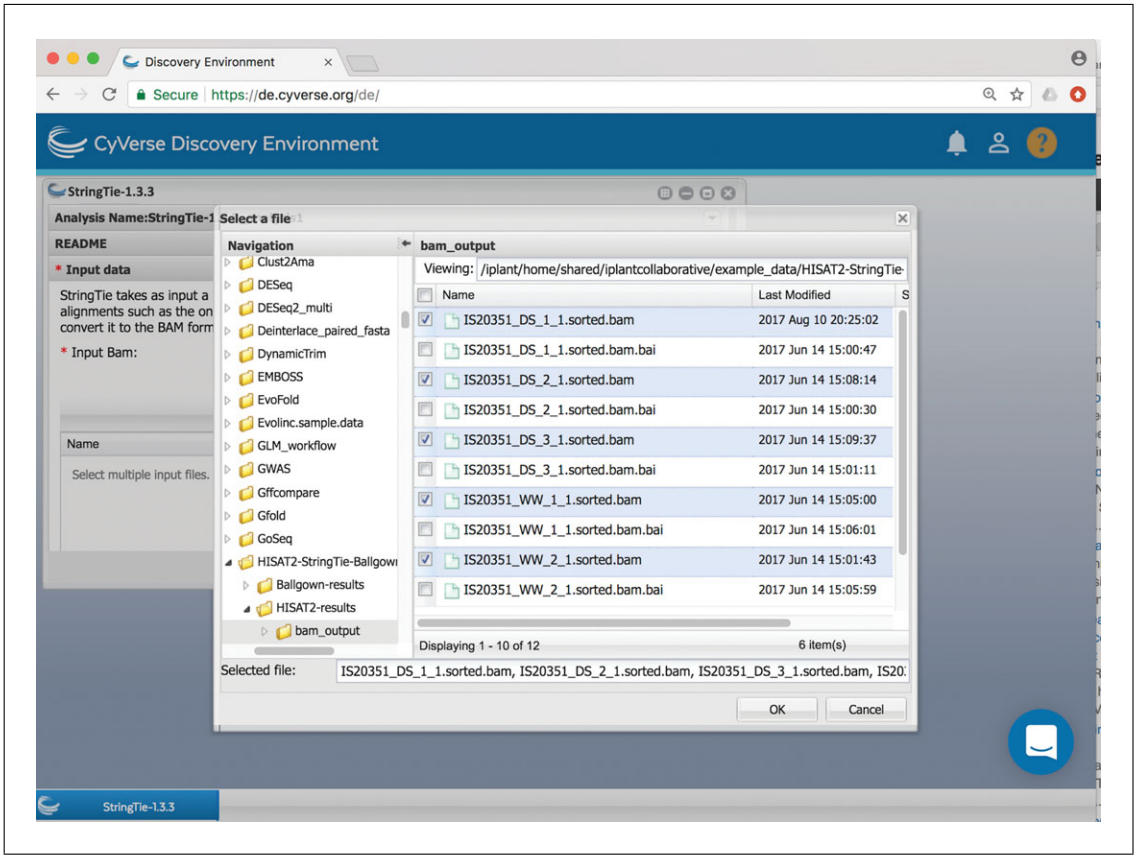

Figure 14 Upload bam files to `StringTie-1.3.3` app. The app accepts multiple bam files for all samples.

29. To run StringTie-1.3.3 again, repeat steps 13 to 14.

30. In the Reference Annotation section, select the consolidated annotation file `merged.out.gtf` generated by the `StringTie-1.3.3_merge` app in step 28, or from the pre-staged results in the CyVerse Data Store. Click on Browse and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *StringTie-1.3.3_merge-results* → *merged.out.gtf*, and click OK (Fig. 23).

    After merged annotation is provided, the pipeline will re-estimate the transcript abundances using the merged structures.

31. Under the Analysis option, you can define the customized parameters for all options; the suggested default values have been set, and clicking on the information button ('i') to the right of an individual parameter will provide further information on that setting. Check both boxes at the bottom to generate table count files needed for differential expression analysis using the Ballgown app (Fig. 24).

32. After making sure that all the input parameter fields have been correctly filled out, click on Launch Analysis to run StringTie-1.3.3.

33. To track the status of your analysis, you can click on the Analyses button.

34. Depending on the number of samples used for transcript assembly, a StringTie-1.3.3 run can take anywhere between 1 to 2 hr.

35. After the job is completed, click on the name of the job in the Analyses window and examine the output files (Fig. 25). The app generates a `StringTie_output` folder that contains the folders described step 20, in addition to read count files for

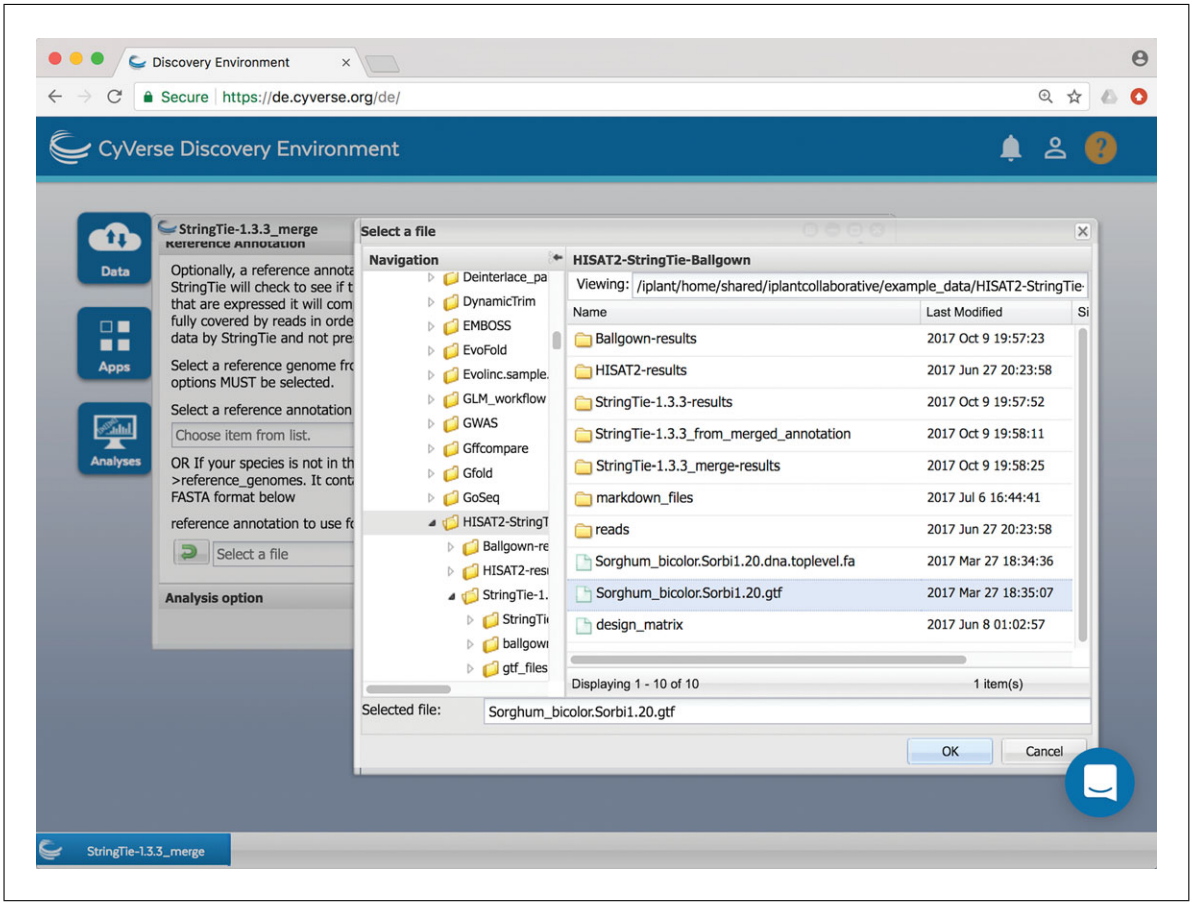

Figure 15 Provide a reference annotation file in GTF format (if available).

each sample to be used with Ballgown. The following count files are generated for each sample under `ballgown_input_files` output folder:

- `$PREFIX.gtf`: transcript assembly in GTF format
- `e_data.ctab`: exon-level expression measurements
- `i_data.ctab`: intron- (i.e., junction-) level expression measurements
- `t_data.ctab`: transcript-level expression measurements
- `e2t.ctab`: table with two columns, e_id and t_id, denoting which exons belong to which transcripts
- `i2t.ctab`: table with two columns, i_id and t_id, denoting which introns belong to which transcripts

Here `$PREFIX` refers to the base name of the bam files. All of these files can be viewed directly from the Discovery Environment or downloaded using the Download drop-down menu in the data window.

Differential expression analysis using Ballgown

Ballgown is an R package that uses abundance data produced by StringTie to perform differential expression analysis at the gene, transcript, exon, or junction level. It can accommodate both time-series and fixed-condition differential expression analyses. Ballgown's statistical methods for differential expression testing uses linear models, but users may wish to use one of the many existing packages like limma, DESeq, or EdgeR, available on CyVerse for differential expression. The default statistical test in Ballgown is a parametric F-test comparing nested linear models. In this example, we will be conducting fixed-condition differential expression analysis. The Ballgown app in

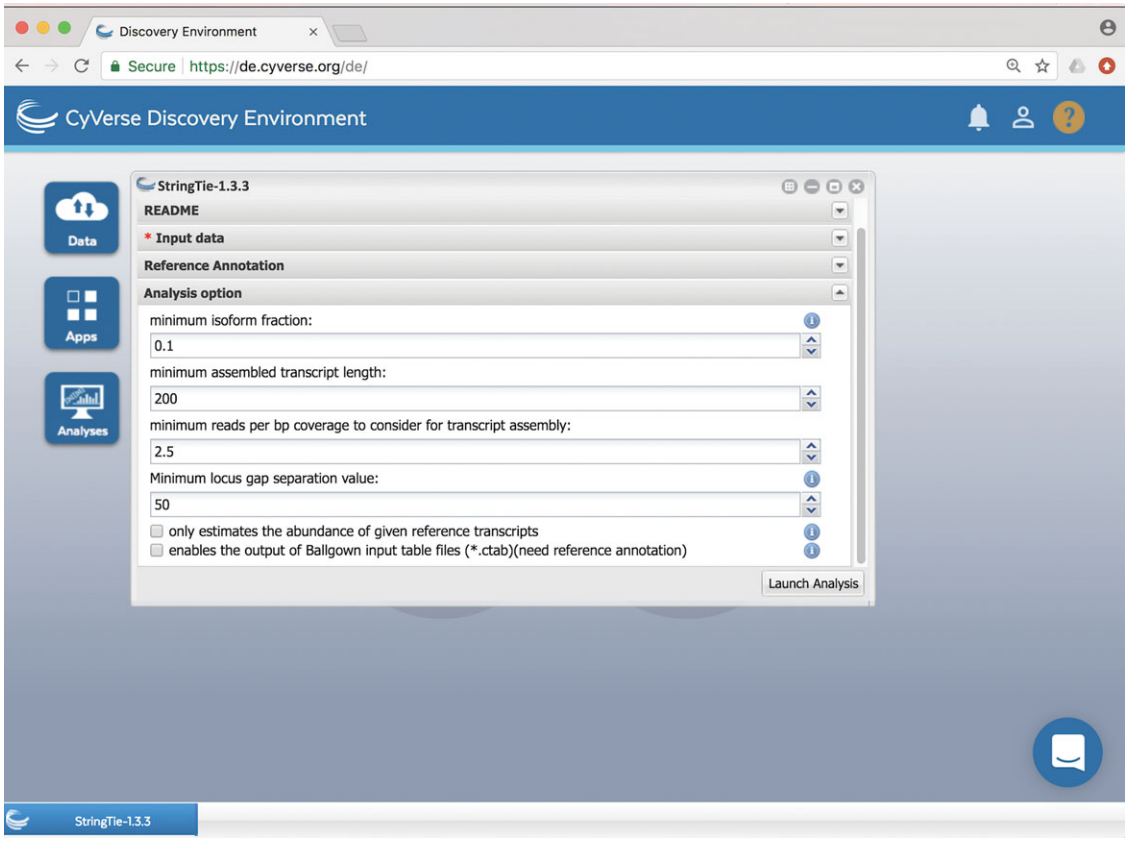

Figure 16 Advanced options for StringTie-1.3.3. For the tutorial, keep all options set to their default values.

DE is a wrapper that takes read count files generated in the previous step and analyzes transcript-level differential expression between conditions.

36. Repeat step 1 to 5, this time searching for Ballgown or navigating to (*Categories* → *Operation* → *Annotation* → *Ballgown*).

37. Once you click on the Input drop-down menu, you will see a section where you can choose your input files. The first input required is an experimental design matrix file which is a text file describing your samples, experimental conditions, replicates used for comparison, or any other experimental covariate (Fig. 26). Click on Browse and select the design matrix file from CyVerse Data store: *Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *design_matrix*.

    Users can customize their own design matrix file based on their experimental design.

38. You are required to provide an experimental covariate of interest on which differential expression will be performed. In our example design matrix, we want to type the experimental covariate as 'condition' for Well watered (WW) Vs Drought Stressed (DS).

    A covariate for the experiment is a user-defined variable on which users wants to perform differential expression analysis: e.g., Condition- salt tolerant Vs sensitive, Treatment- hot Vs cold (etc.).

    The experimental covariate provided by the user should match the column name in the design matrix file, and the folder names for the read count files should match the ID

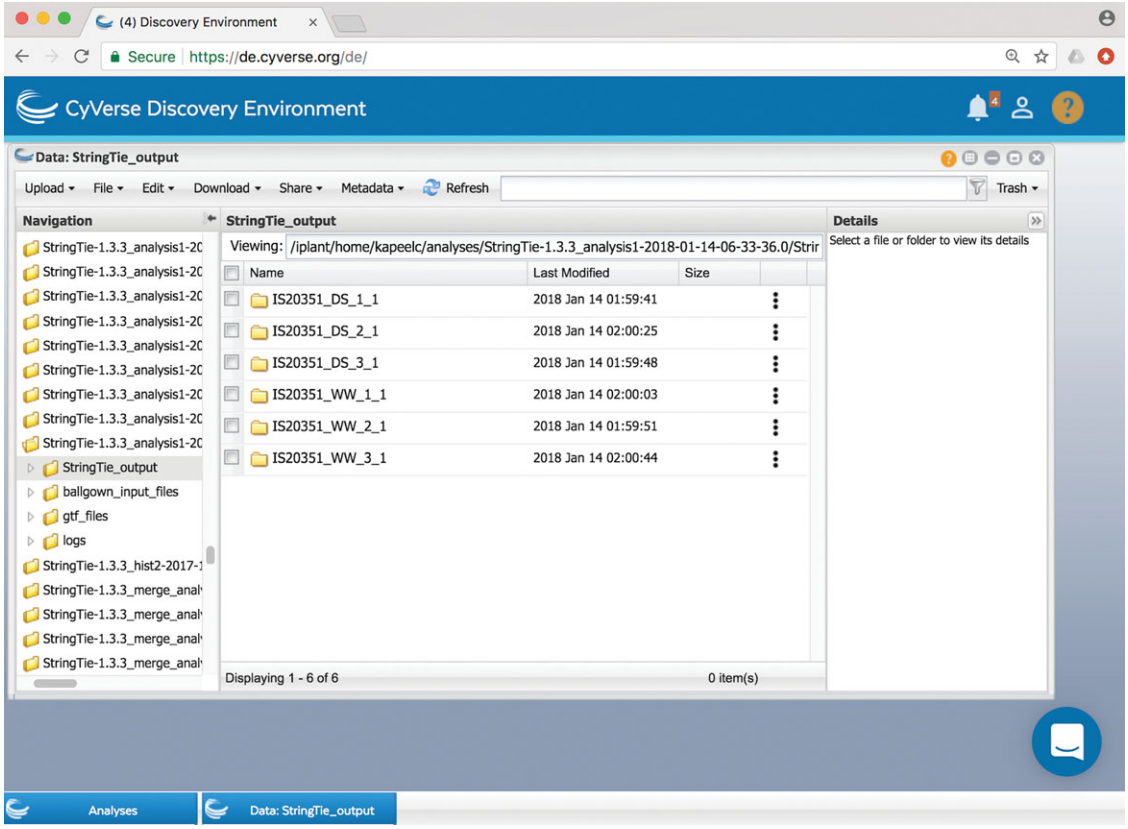

Figure 17 StringTie output folder with `.gtf` files for each file.

names in the design matrix files. This app only allows for pairwise comparisons; for multi-group and time series comparisons, Ballgown can be run from the command line within the R environment.

39. Finally, provide the read counts generated for Ballgown using the StringTie.1.3.3 app (*Community Data* → *iplantcollaborative* → *example_data* → *HISAT2-StringTie-Ballgown* → *StringTie-1.3.3_from_merged_annotation* → *ballgown_input_files*) (Fig. 27).

40. After making sure that all the input parameter fields have been correctly filled out, click Launch Analysis to run Ballgown.

41. To track the status of the analysis, click on the Analyses button.

42. Depending on the number of samples used for transcript assembly, a Ballgown run can take anywhere from 5 to 10 min.

43. After the job is completed, click on the name of the job in the Analyses window and examine the output (Fig. 28). The app generates the following files:

    - `Rplots.pdf`: boxplot of FPKM distribution of each sample
    - `results_gene.tsv`: gene-level differential expression with no filtering
    - `results_gene_filter.sig.tsv`: genes with p value < 0.05
    - `results_gene_filter.tsv`: Filter low-abundance genes
    - `results_trans.tsv`: transcript-level differential expression with no filtering
    - `results_trans_filter.sig.tsv`: transcripts with p value < 0.05
    - `results_trans_filter.tsv`: Filter low-abundance transcripts.

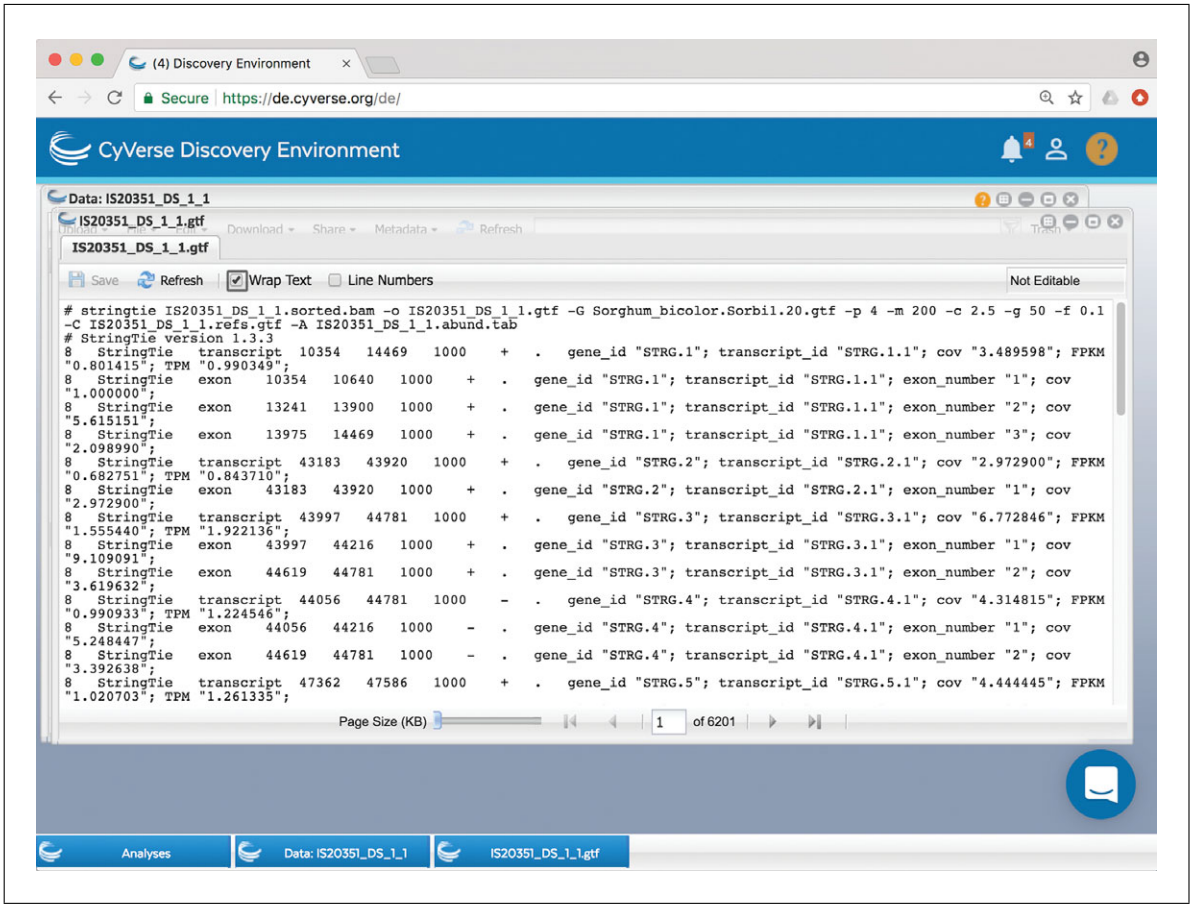

Figure 18 Nine-column gtf file. The last column contains FPKM, TPM, and coverage values defining the expression of a transcript.

BASIC PROTOCOL 2

PSEUDO-ALIGNMENT-BASED PROTOCOL USING KALLISTO AND SLEUTH

If the goal of the experiment is quantification of transcripts or genes, newer pseudo-alignment-based approaches offer a faster alternative to the traditional Tuxedo protocol. Instead of mapping reads to a reference, pseudo-alignment protocols identify a potential for read to arise from a transcript, thereby skipping the alignment step and decreasing run time. The Kallisto-Sleuth pipeline comprises two steps: (a) Kallisto for indexing and quantification; and b) Sleuth for analysis of isoform-level differential expression. This simplifies the analysis from the user's point of view because there are no large alignment files needed for quantification of isoforms.

The protocol below describes Kallisto quantification using the Kallisto and Sleuth apps in DE. Kallisto is primarily meant for quantification of transcript sequences and does not perform transcript assembly or find novel isoforms, but you can run Kallisto for quantification on assembled transcripts from other programs. In addition to performing statistical tests for differential expression, Sleuth also offers interactive visualization of results using the R *Shiny* package, which the users can run in an Rstudio environment, this makes it easy for users to interpret the results.

Necessary Resources

Hardware

A computer with Internet access

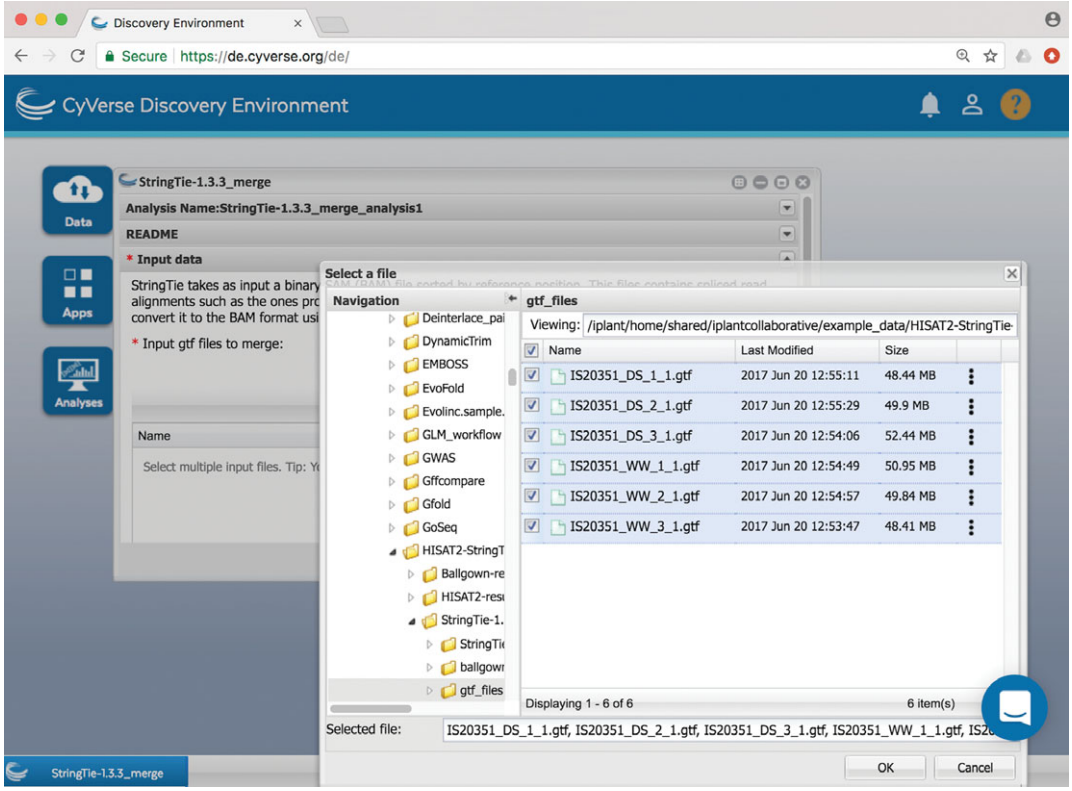

Figure 19 Assembled gtf files for merging with `StringTie-1.3.3_merge`.

Software

An up-to-date Web browser, such as Internet Explorer (*https://www.microsoft.com/ie/*), Firefox (*https://www.mozilla.com/*), Chrome (*https://www.google.com/chrome*), or Safari (*https://www.apple.com/safari/*). JavaScript must be enabled in the Web browser.

Files

Reference transcriptome in FASTA format. The `Sorghum_bicolor.Sorbi1.20.cdna.all.fa`file can be obtained from the Gramene project FTP site: *ftp://ftp.ensemblgenomes.org/pub/plants/release-20/fasta/sorghum_bicolor/cdna/Sorghum_bicolor.Sorbi1.20.cdna.all.fa.gz*. File should be unzipped before using in the app.

RNA reads in FASTQ format (single-end or paired-end); the read file can also be passed in compressed form in gzip `(.gz)` format

Input and output files for Basic Protocol 2 are staged on the CyVerse Data Store, accessible via the Discovery Environment. Once you log in to the Discovery Environment, click on the Data tab and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *Kallisto_Sleuth* (Fig. 29)

1. Using your CyVerse credentials, log in to the Discovery Environment of the CyVerse Web site: *https://de.cyverse.org* (Fig. 3).

   To access the Discovery Environment, you must first register at https://user.cyverse.org/register.

Kallisto indexing and quantification

2. Once you are logged in, click the Apps button on the left side of the window (Fig. 4) to get the full list of apps available on CyVerse.

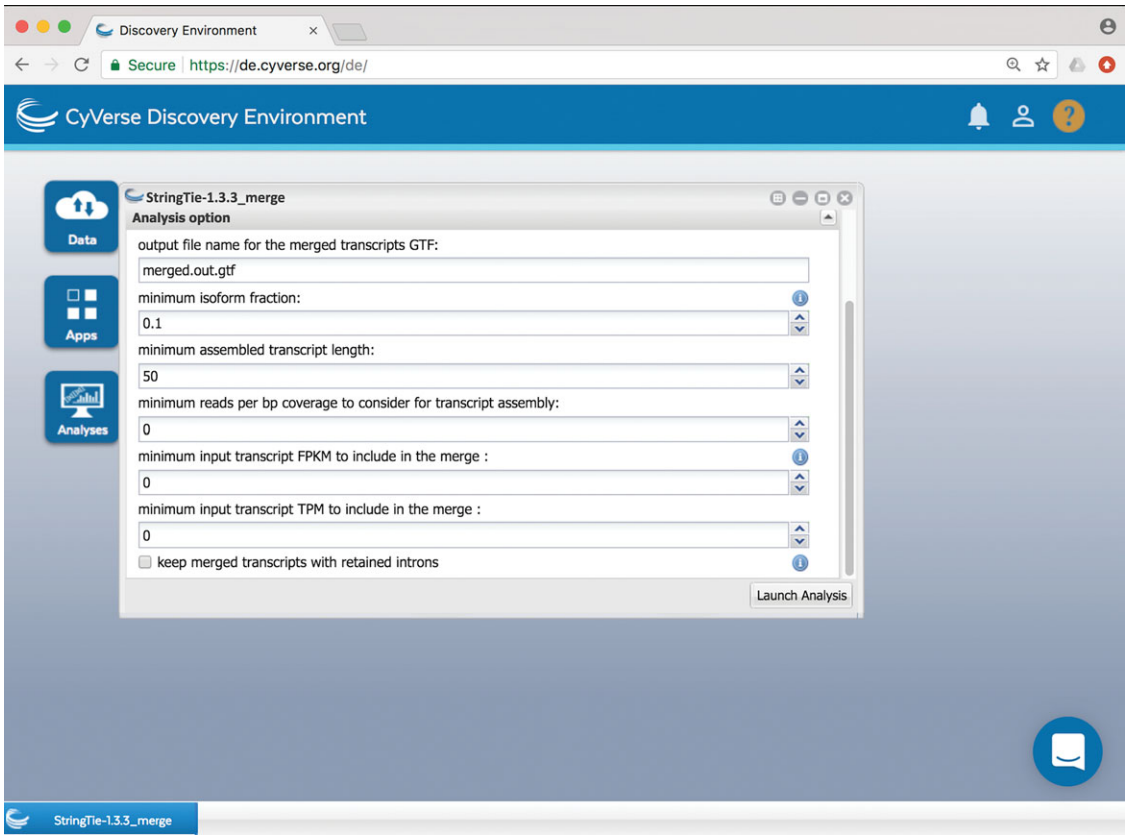

Figure 20 Provide a reference annotation file in GTF format (if available).

3. Either search for Kallisto-v.0.43.1 or navigate around the list of apps in the left navigation column (*Categories Operation → Generation → Alignment → Kallisto-v.0.43.1*) to find "Kallisto-v.0.43.1" (Fig. 30).

   Some other versions of Kallisto seen are built by other Cyverse users. Cyverse provides users to build and customize a GUI for their own apps or build on an existing app's GUI. Some of the apps are built with Agave API to run on High Performance Cluster. The advantage of using Kallisto-v.0.43.1 is that it allows users to run Kallisto on multiple samples in a single job submission

4. Once you click on this app, you will see another pop-up window where you will run your analysis (Fig. 31).

5. In the first section of the app, give a name to your session, write a comment about this particular analysis, and set the name and location of directory where all of the output files will be deposited.

6. Once you click on the Inputs drop-down menu, you will see a section where you can choose your input files. Here, provide a transcriptome cDNA or EST sequence file in FASTA format, either by uploading or by selecting from the Discovery Environment by clicking on the Browse button. Both single-end and paired-end sequencing reads in FASTQ format can be used for the analysis, which you must also upload and provide a location for (Fig. 32).

   Kallisto uses a transcriptome file instead of genome sequence file for pseudo-alignment and quantification. The app allows user to upload multiple read files for all the samples. If using paired-end reads, load the read1 files under FASTQ Files (Read 1) and read2

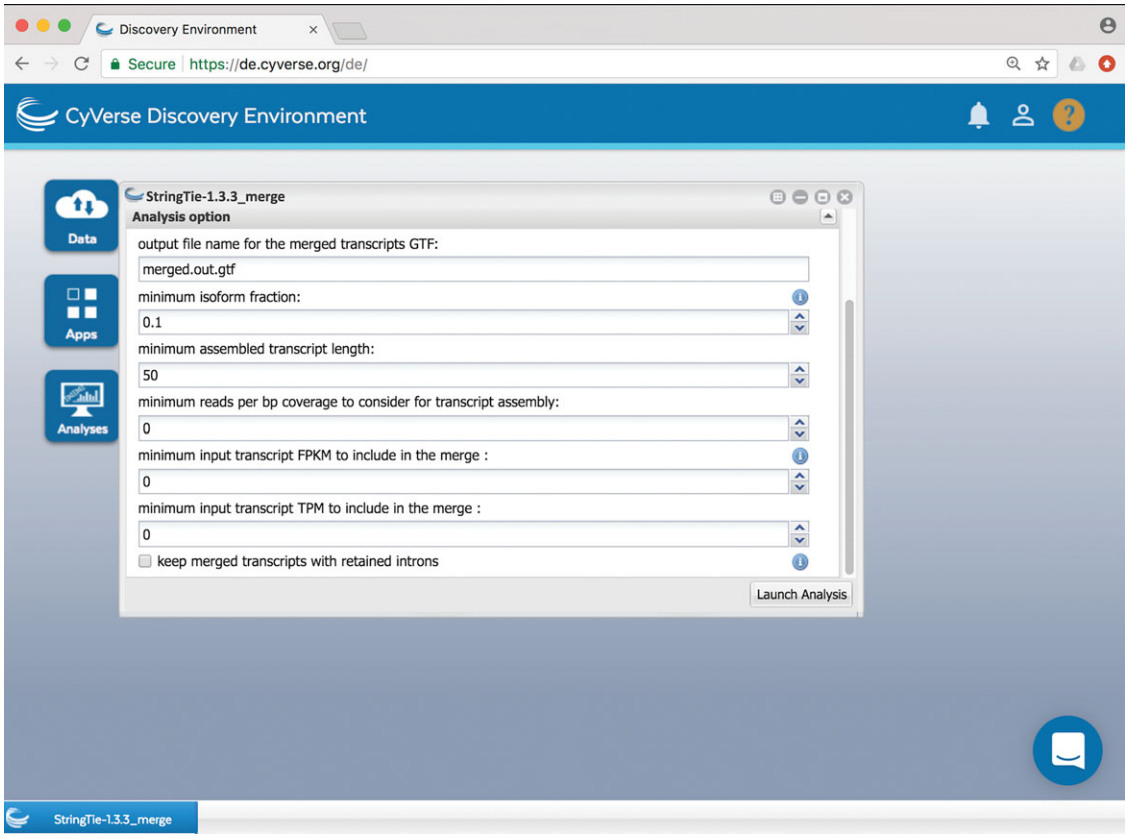

Figure 21 Advance options for `StringTie-1.3.3_merge`. Keep all options at the default values.

files under FASTQ Files (Read 2). For single-end reads, load all the read1 files under FASTQ Files (Read 1) and leave the FASTQ Files (Read 2) empty.

7. In our example, we will use *Sorghum bicolor* v1.2 as the reference transcriptome, and paired-end read files. To upload the reference transcriptome, click on Browse and navigate to *Community Data* → *iplantcollaborative* → *example_data* → *Kallisto_Sleuth*. Select the file *Sorghum_bicolor.Sorbi1.20.cdna.all.fa* and click OK. Similarly, load the read 1 and read 2 files separately for all samples: *Community Data* → *iplantcollaborative* → *example_data* → *Kallisto_Sleuth* → *reads*. Select 'paired' for library type paired. To upload read1 files, click on the read1 folder and select all the read1 files by checking the box before the file names. The app allows multiple file uploads. The user can select all files at once by checking the box before the Name column (Fig. 10) and clicking OK. Similarly, upload the read2 files under FASTQ Files (Read 2), and use files in the read2 folder. Make sure that the order of the files in both read1 and read2 inputs is the same.

   Users can upload their own data using the upload feature described at https://wiki.cyverse.org/wiki/display/DEmanual/Uploading+and+Importing+Data+Items+Within+the+DE, which contains instructions that allow users to bring in their own RNA-seq data in FASTQ format and reference genome in FASTA format and upload it to data store. Once on the data store, the user can click on browse option with the Kallisto-v.0.43.1 app to provide the input files.

   Simple upload from the desktop allows users to upload files of maximum size 1.9 Gb. For uploading files of larger size, and for bulk upload, see Troubleshooting section. There is a 100-Gb disk allocation limit for a CyVerse user account; for additional disk space the user can submit a request for data allocation increase at https://user.cyverse.org/forms/2 with the justification for the allocation increase.

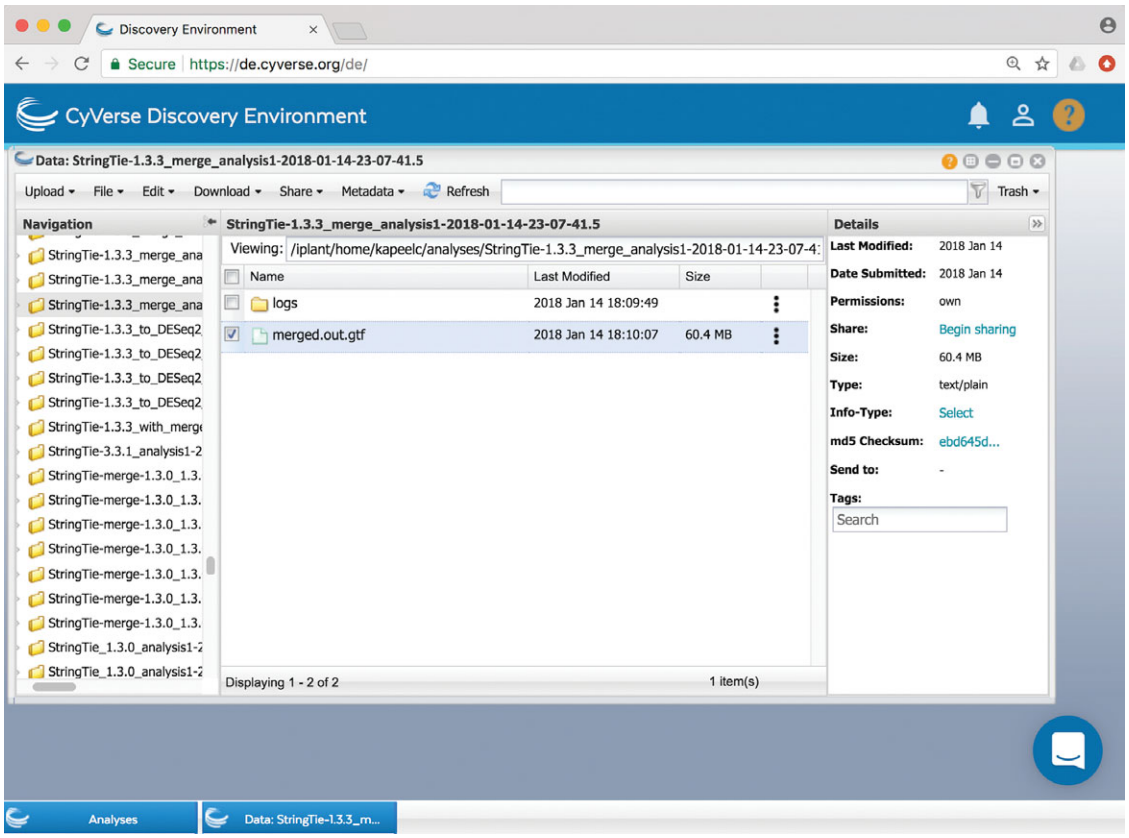

Figure 22 Consolidated merged annotation output file from `StringTie-1.3.3_merge` app.

8. Click on the Options drop-down menu to set all parameters for the run. For your convenience, the default values have been set in each field, and clicking on the information button ('i') to the right of an individual parameter will provide further information on that setting. The first option the user can provide is the number of bootstrap iterations. Bootstrap provides a measure of the accuracy of the quantification by random resampling with replacement from the read data to determine uncertainty estimates. Choosing a higher number for bootstraps is preferred, as it allows one to obtain a better estimate of the technical variance, which is very useful when testing for differentially expressed transcripts with the Sleuth tool. There is no community standard, but, in general, a good confidence value ranges from 50 to 100. In this example, we will set the bootstrap to 60. A larger value will result in longer run times. If using single-end data, the user must provide an estimate of the average fragment length. Typical Illumina libraries produce fragment lengths ranging from 180 to 200 bp, but it is best to determine this from library quantification using an instrument such as an Agilent Bioanalyzer. For paired-end reads, the average fragment length can be directly estimated from the reads. The user can save the pseudo-alignments to the transcriptome to a BAM binary format that could be used for visualization in a genome browser. The user needs to check one of the last two parameters if the input reads are strand-specific, but leave them unchecked if the reads are unstranded.

   By default, Kallisto runs zero bootstrap iterations. If you do not plan to run Sleuth for differential expression analysis, this is okay. However, if you plan to run Sleuth, you must provide a nonzero number of bootstraps.

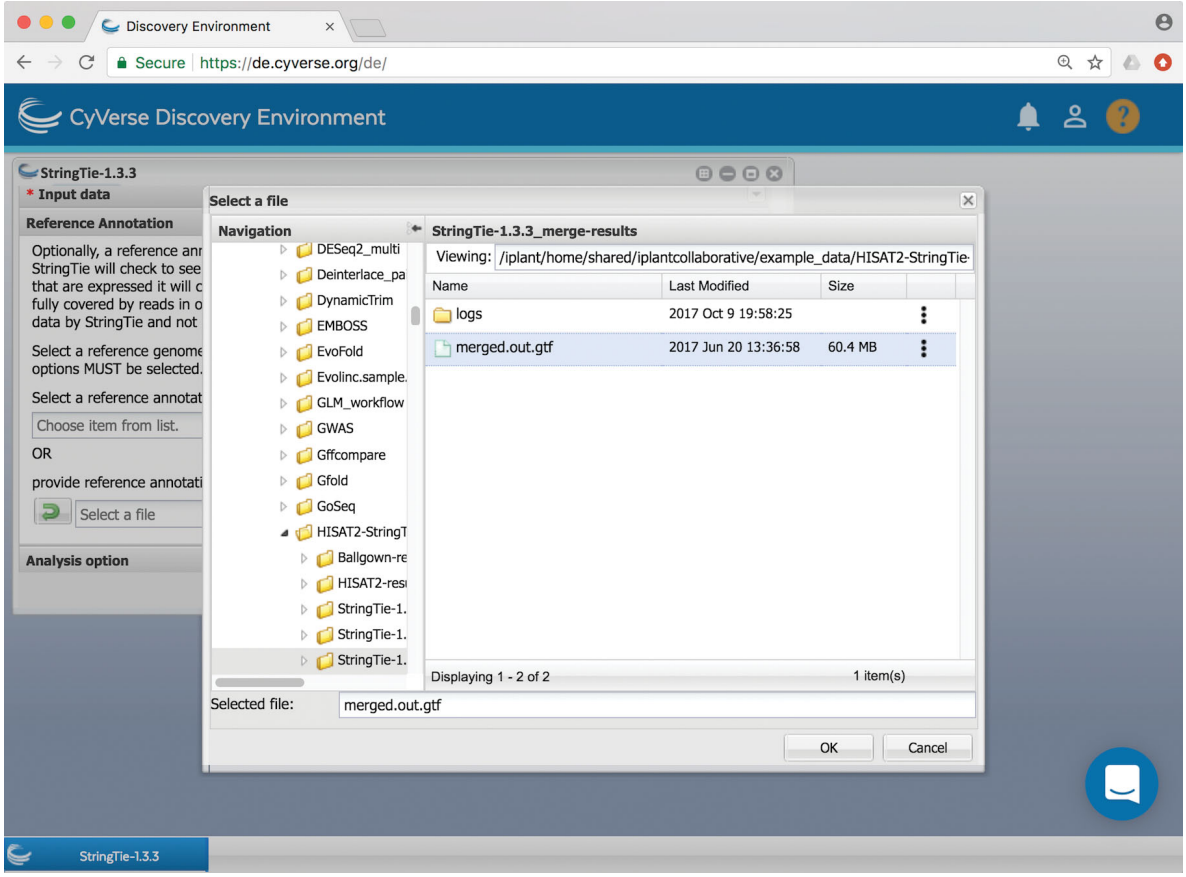

Figure 23 Selecting consolidated merged annotation for re-run of `StringTie-1.3.3` app.

9. After making sure that all the input parameter fields have been correctly filled out, click on Launch Analysis at the bottom right corner of the app window to run Kallisto-v.0.43.1 (Fig. 33).

   Once the app has launched, you will receive a prompt alerting you that your job has been submitted. As a default, you will receive an e-mail notification when the status of the job changes to Complete.

10. To track the status of the analysis, click on the Analyses button in the left panel to get the list of analyses that have run in the Discovery Environment, and check the status by looking at the Status section. Alternatively, you can click on the bell-shaped icon in the top-right panel to see the status of your most recently launched jobs (Fig. 34).

11. Depending on the size of the transcriptome and sequencing reads, it may take up 1 to 2 hr for the analysis to run to completion. Once the job is complete, you will receive an e-mail notification saying that the status of your run has changed from 'running' to 'completed'.

12. After the job is completed, click on the name of the job from the Analyses window and examine the output files. All of the files generated from the analyses are permanently available to the user. The app creates an output directory containing the following output files (Fig. 35):

- `run_info.json`: some high-level information about the run, including the command and versions of Kallisto used to generate the output.

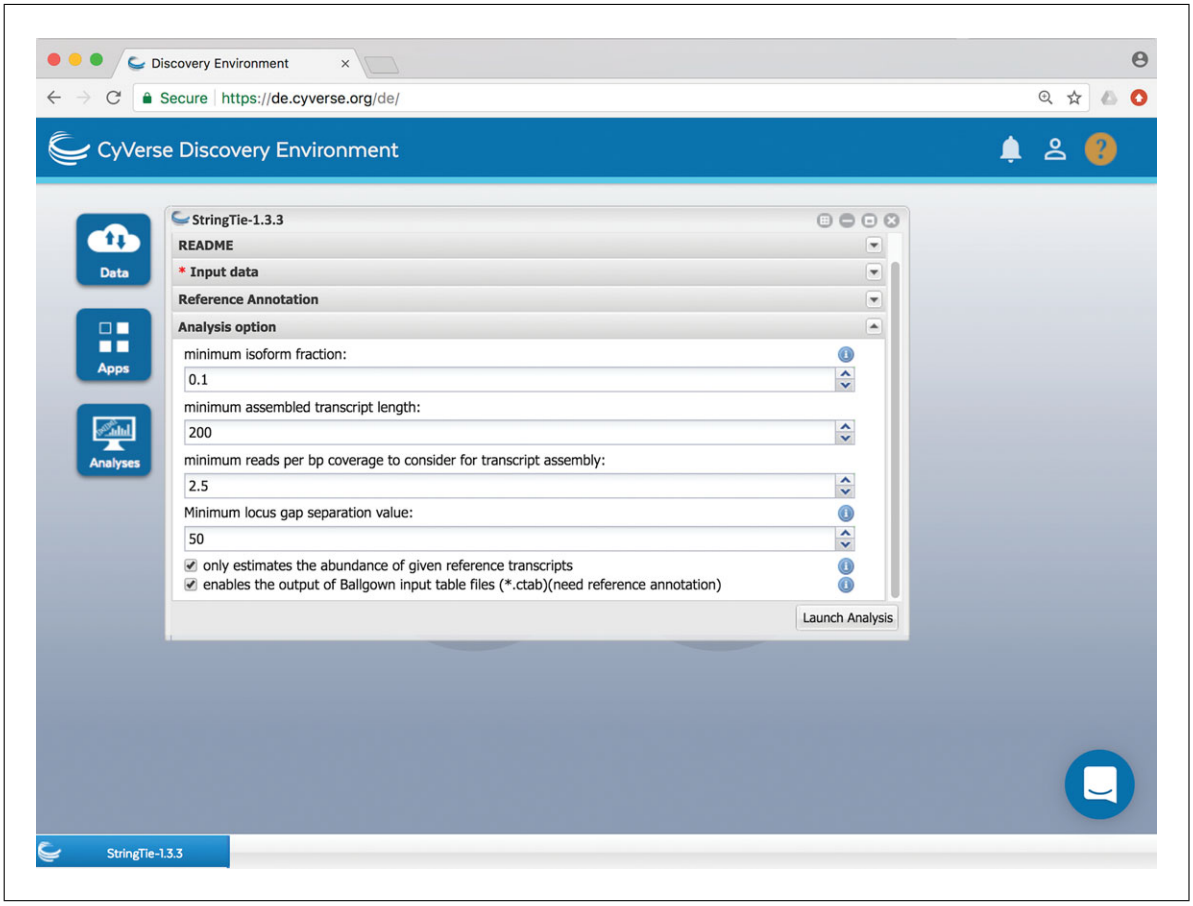

Figure 24 Enable options to create Ballgown table count files.

- `abundance.tsv`: a plain text file with transcript-level abundance estimates. This file can be easily read into R or any other statistical language [e.g., `read.table('abundance.tsv')`].
- `abundance.h5`: an HDF5 file containing all of the quantification information, including bootstraps and other auxiliary information from the run. This file is read by Sleuth.

Differential expression analysis using Sleuth

Sleuth is an R package that uses transcript abundance estimates produced by Kallisto to identify biological differences in transcript expression. Sleuth implements a general linear model as its statistical model and applies the likelihood ratio test, where the full model contains the labels for samples and the reduced model ignores the label. In this example, we will be conducting fixed-condition differential expression analysis. The Sleuth app in DE is a wrapper that takes bootstrapped transcript abundances generated by Kallisto and analyzes transcript-level differential expression between conditions. To use the Sleuth app, RNA-seq data must first be quantified with Kallisto app as described in steps 2 to 12 of Basic Protocol 2.

13. Repeat steps 1 to 5, this time searching for Sleuth or navigating *Categories* → *Operation* → *Annotation* → *Sleuth*) to Sleuth.

14. Once you click on the Input drop-down menu, you will see a section where you can choose your input files. The first input required is an experimental design matrix file, which is a text file describing your samples, experimental conditions, replicates, and other experimental factors used in your experiment (Figs. 36, 37). Select the design

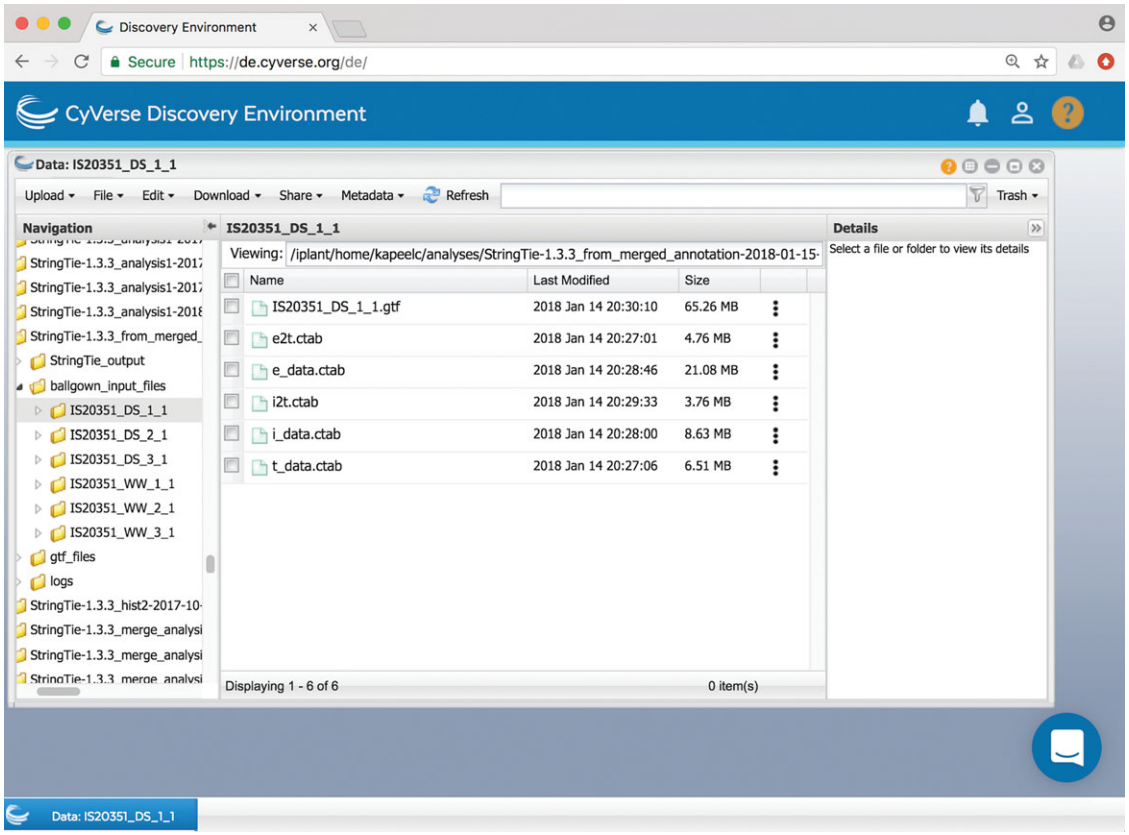

Figure 25 Read count generated for Ballgown differential expression analysis.

matrix file from the CyVerse Data store (*Community Data* → *iplantcollaborative* → *example_data* → *Kallisto_Sleuth* → *design_matrix*). Make sure the first column name is 'sample'. The next two options require the user to provide formulae for the full and reduced model. The full model contains the label for the samples, while the reduced model ignores the labels. In our experimental data, since we want to perform differential expression based on condition, we use '~condition' as our full model and for reduced model we pass '~1', which represents the intercept variable in the Sleuth statistical model. Users can define their own formulae for full and reduced model based on the experimental design. Make sure when defining the formulae to prefix '~'. For highlighting the covariate in the Sleuth plots, the user needs to provide an experimental covariate of interest (in this case condition) on which differential expression will be performed (Fig. 38). Finally, provide the abundance estimates output folder from the Kallisto app (*Community Data* → *iplantcollaborative* → *example_data* → *Kallisto_Sleuth* → *kallisto_quant_output*) (Fig. 39).

Users can customize their own design matrix file based on the comparison that needs to be performed. The experimental covariate provided by the user should have a match in the column names of the design matrix file, and the folder names for the read count files should match the sample column names in the design matrix files. This app only allows for pairwise comparisons; for multi-group and time-series comparisons, Sleuth can be run from the command line within the R environment.

15. After making sure that all the input parameter fields have been correctly filled out, click Launch Analysis to run Sleuth.

16. To track the status of the analysis, click on the Analyses button.

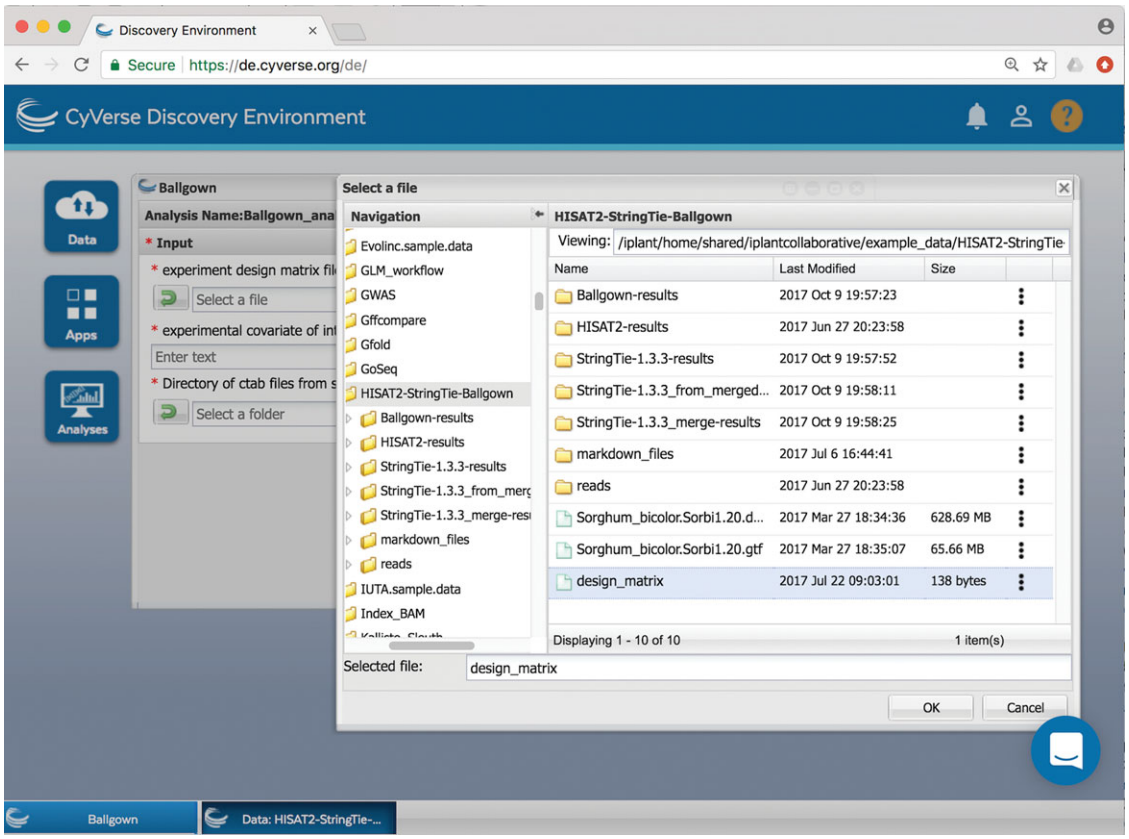

Figure 26 Design matrix file describing the experimental conditions on which differential expression analysis will be performed by the Ballgown app.

17. Depending on the number of samples used for transcript assembly, a Sleuth run can take anywhere from 5 to 10 min.

18. After the job is completed, click on the name of the job in the 'Analyses' window and examine the output (Fig. 40). The app generates the following files:

    - `results_trans.tsv`: Sleuth result table with below details
      - `target_id`: transcript name
      - `pval`: p value
      - `qval`: FDR-adjusted p value using Benjamini-Hochberg
      - `mean_obs`: the mean of the observations. This is used for the smoothing.
      - `var_obs`: the variance of the observations
      - `tech_var`: the technical variance from the bootstraps
      - `sigma_sq`: the raw estimator of the variance once the `tech_var` has been removed
      - `smooth_sigma_sq`: the smooth regression fit for the shrinkage estimation
      - `final_sigma_sq - max(sigma_sq, smooth_sigma_sq)`: this is the one used for covariance estimation of beta (in addition to tech_var)
    - `results_trans.sig.tsv`: filter significantly differentially expressed transcripts with qval ≤ 0.5
    - `sleuth_obj`: Sleuth object saved that can be used with the `sleuth_deploy` function for explorative analysis of results for the Sleuth Shiny application.
    - `pca.jpg`: Principal Component Analysis plots for first and second component, useful to see how the samples the cluster and measure variance between samples

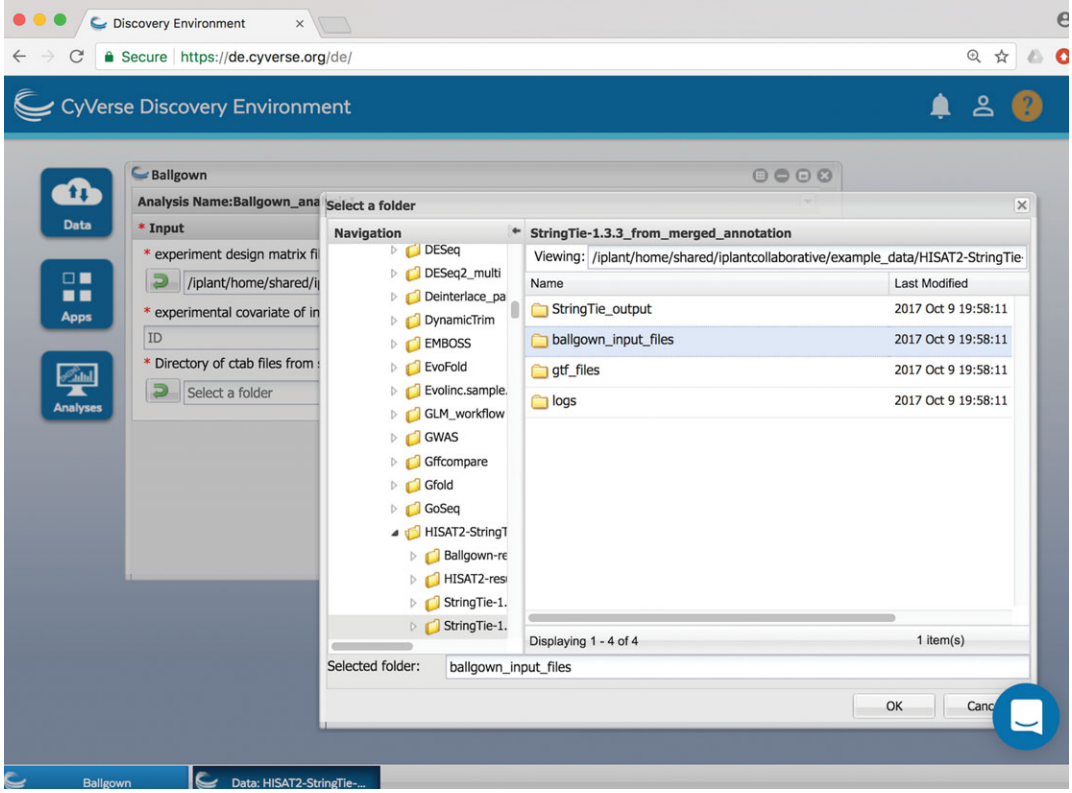

Figure 27 Ballgown input files containing read counts files for each sample.

- `mean_variance.jpg`: Plot of abundance versus square root of standard deviation which is used for shrinkage estimation. The blue dots are in the interquartile range and the red curve is the fit used by sleuth
- `density.jpg`: The count distributions for each sample grouped by the experimental covariate provided

GUIDELINES FOR UNDERSTANDING RESULTS

Basic Protocol 1

This protocol was designed to show users how to run reference-guided assembly-based RNA-seq analysis using the CyVerse Discovery Environment platform. The protocol describes the newly updated Tuxedo protocol using the HISAT2, StringTie, StringTie-merge, and Ballgown apps integrated into the DE platform. Each app can be run independently, allowing users to upload multiple input files. The protocol describes how users can upload their data for each app in the browser, or use pre-staged data on the CyVerse data store.

This particular protocol requires that the user provide a reference genome and paired- or single-end RNA-seq reads from one or more samples. HISAT2 creates a genome index and aligns the reads to the reference sequence simultaneously, creating a BAM alignment index file containing mapping of reads against the reference (steps 1 to 12). HISAT2, by default, outputs a human-readable SAM-format alignment file, but the HISAT2 app converts the output from SAM to BAM for convenience because the SAM files occupy a great deal of disk space, whereas bam files are easily portable and visualized by other downstream applications.

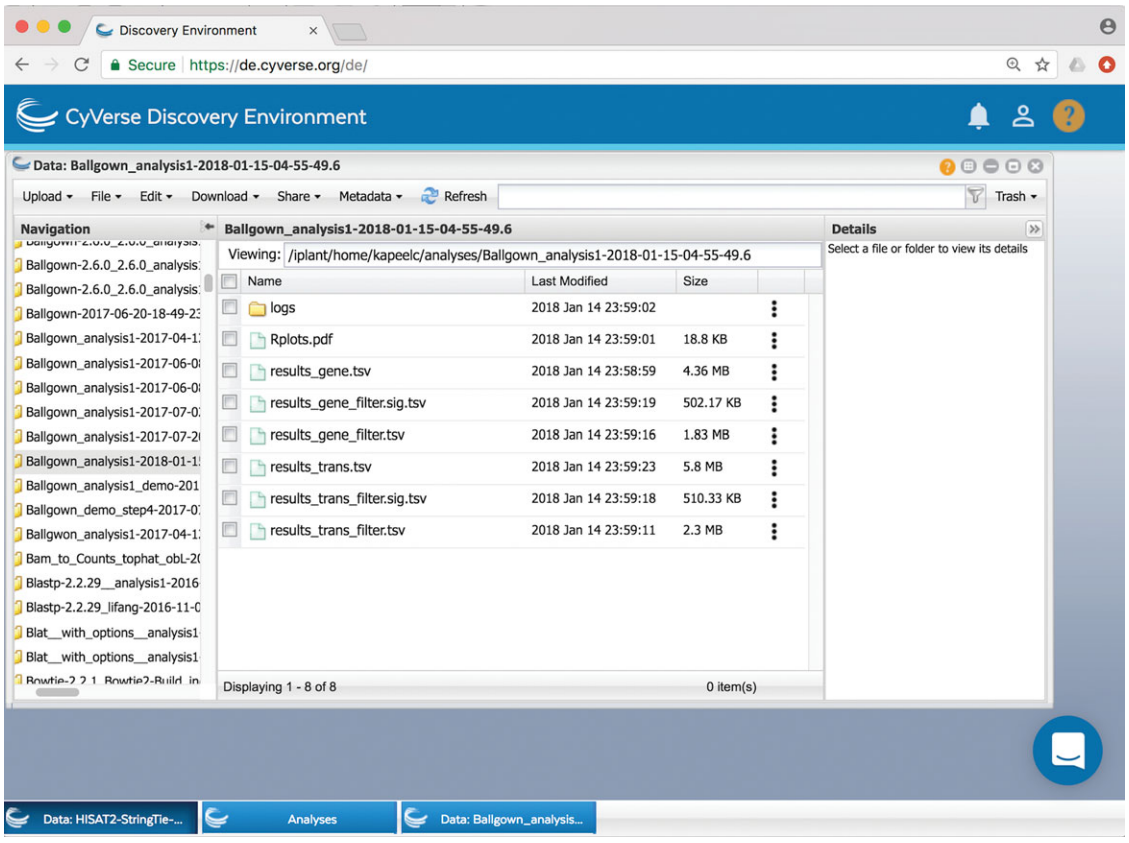

Figure 28 Ballgown output files.

Furthermore, the protocol shows how to construct transcript structures using the StringTie app (steps 13 to 20), where the user provides the BAM alignment files (obtained from HISAT2 at step 12) as input, and then outputs a GTF format assembly file. The assembled transcripts GTF file for each sample contain Fragments Per Kilobase of transcript per Million mapped reads (FPKM), Transcripts Per Million (TPM), and per-base coverage values, which give an overall expression of the constructed transcript as well as the mapping coordinates of the genomic feature. Some transcripts will have higher or lower expression values, which users can filter based on their experimental needs. A recommended practice is to filter low-expressed transcripts (defined as those with FPKM < 1), but this can sometime remove transcripts that are actually expressed; this is especially the case with transcription factors, which are expressed at low levels. By default, the StringTie app keeps all transcripts, even those with FPKM $= 0$, TPM $= 0$, or no read coverage support.

It is important to consolidate these assembled transcript structures with the reference annotation file, if provided by the user, using the StringTie-merge app (steps 21 to 28). The merging will remove any redundant transcripts, and merge transcripts to provide complete transcript structures that can better estimate transcript abundances and facilitate the downstream calculation of differential expression levels for all transcripts among the experimental conditions. Output is a merged GTF file with all merged gene models, but without any numeric values for coverage, FPKM, or TPM. With this merged GTF, the StringTie app can be used to re-estimate transcript abundances by running it again with options selected to create the Ballgown input table files (steps 29 to 35).

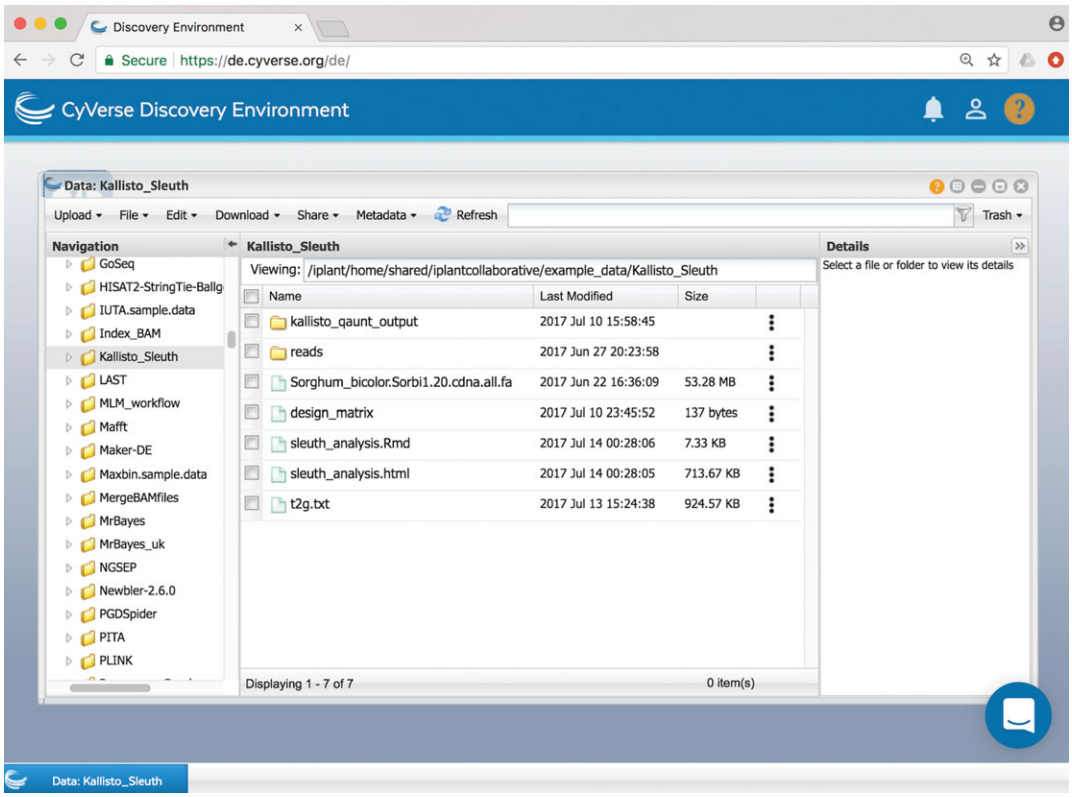

Figure 29 Staged input and output files for Basic Protocol 2.

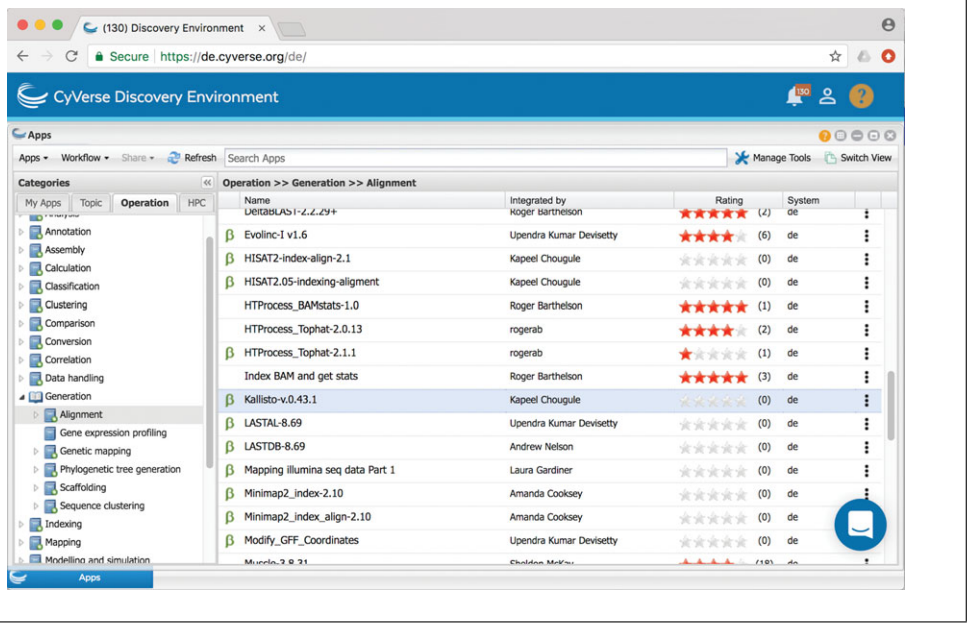

Figure 30 Navigate to find `Kallisto-v.0.43.1` in the Discovery Environment.

Finally, the protocol demonstrates how to use the Ballgown app to compare differential expression in a drought-sensitive sorghum line under drought stress (DS) and well-watered (WW) conditions. The Ballgown app outputs both differentially expressed genes and transcripts in separate files under both conditions, along with separate output files with significantly differentially expressed genes with cut-off p value < 0.05 (steps 36

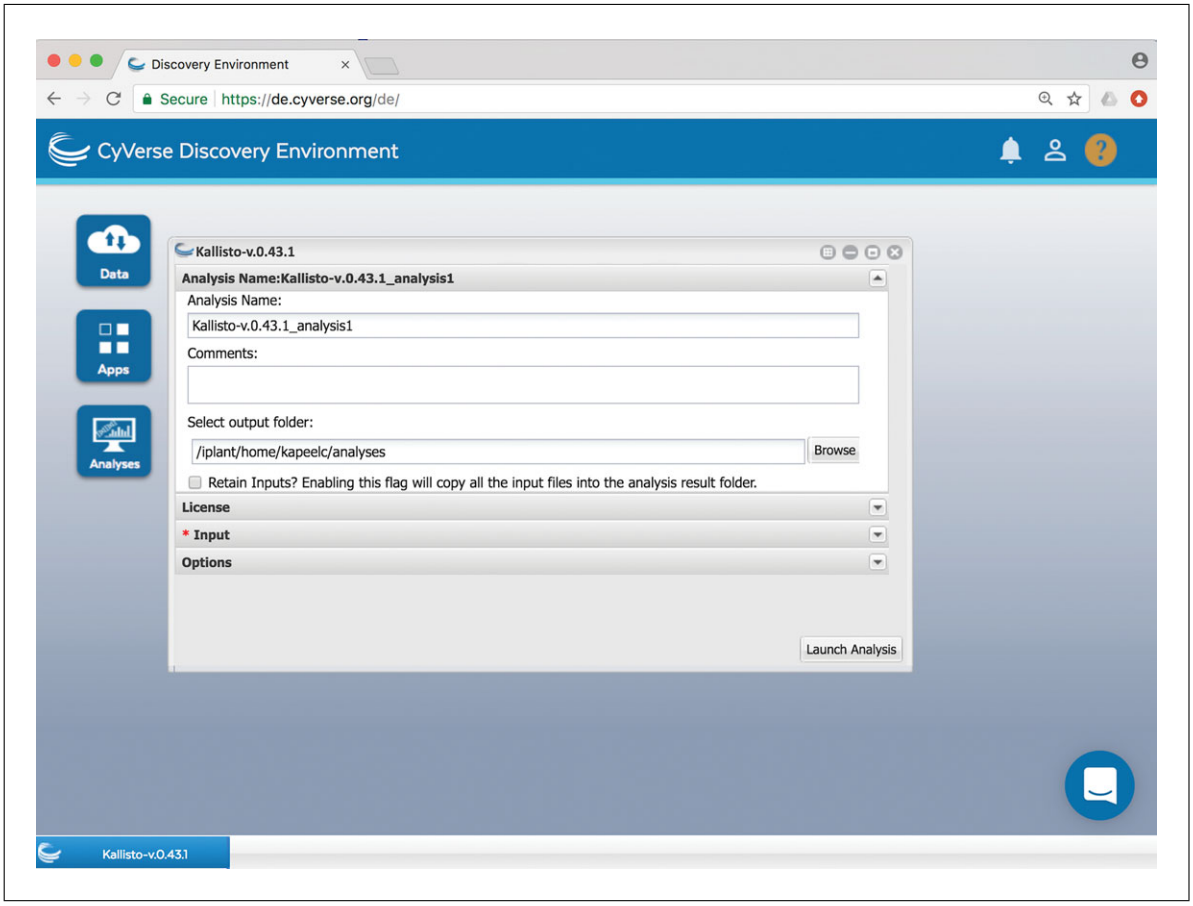

Figure 31 Pop-up window of `Kallisto-v.0.43.1` app in the Discovery Environment.

to 43). To obtain meaningful and reliable results using this protocol, it is recommended that the user have at least three experimental replicates for each sample. This protocol identifies both novel and reference genes that are differentially expressed. The Ballgown app can similarly be used for identifying differentially expressed transcripts or genes given an experimental design matrix file.

Basic Protocol 2

This protocol was designed to show users how to run pseudo-alignment-based RNA-seq analysis using the CyVerse Discovery Environment. The protocol describes using the Kallisto and Sleuth app in the Discovery Environment to pseudoalign paired or single-end RNA-seq reads from one or more samples against a reference transcriptome in order to generate transcript abundances for each sample and to detect differentially expressed transcripts.

This app creates a transcriptome index and pseudoaligns the reads at the same time (steps 1 to 12). It provides options to bootstrap the pseudo-alignment process; in addition, because of the large amount of data that may be produced when the number of bootstrap samples is high, Kallisto outputs bootstrap results in HDF5 format. The `h5dump` command can be used afterwards to convert this output to plain text; however, it is most convenient to analyze bootstrap results with Sleuth. The transcript abundances for each sample are measured in read counts and TPM The bootstrapped transcript abundance are used in Sleuth app to compare differential expression in a drought-sensitive sorghum line under drought stress (DS) and well-watered (WW) conditions. The Sleuth app outputs all differentially expressed transcripts under both conditions, along with a separate output file with significantly differentially expressed genes with cut-off of q value < 0.05 (steps

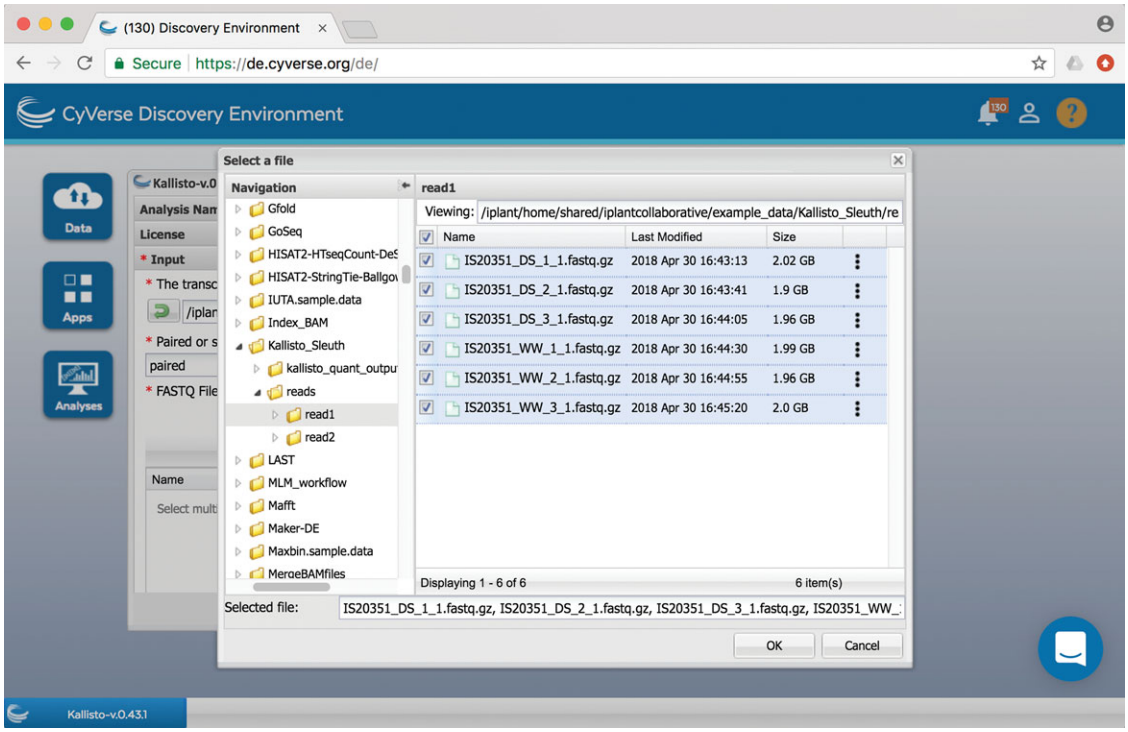

Figure 32 Fields to specify inputs for `Kallisto-v.0.43.1` run. Required input files are the reference transcriptome in FASTA format and either one read file in FASTQ or compressed format (`.gz`) for single-end reads or two read files in the same format for paired-end reads.

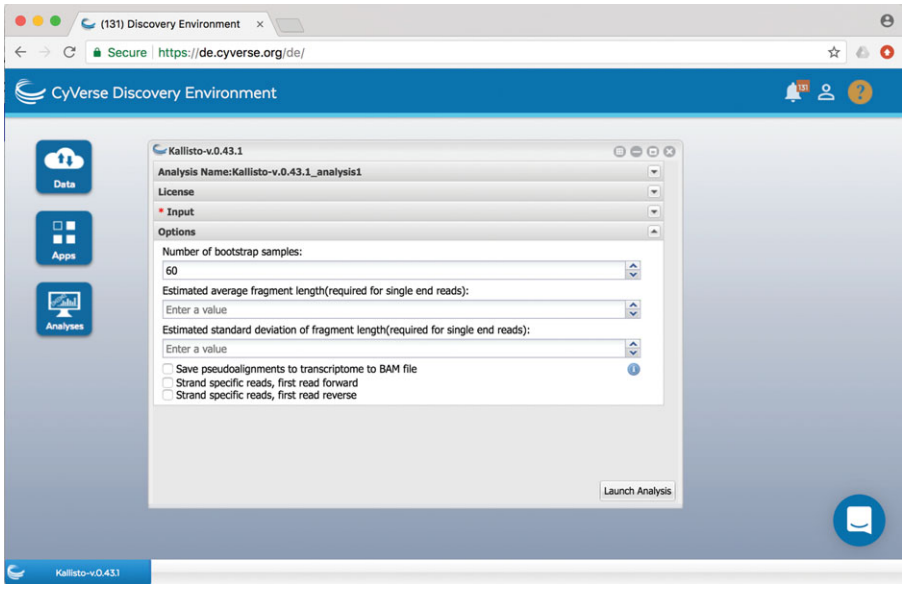

Figure 33 Other options for the `Kallisto-v.0.43.1` app.

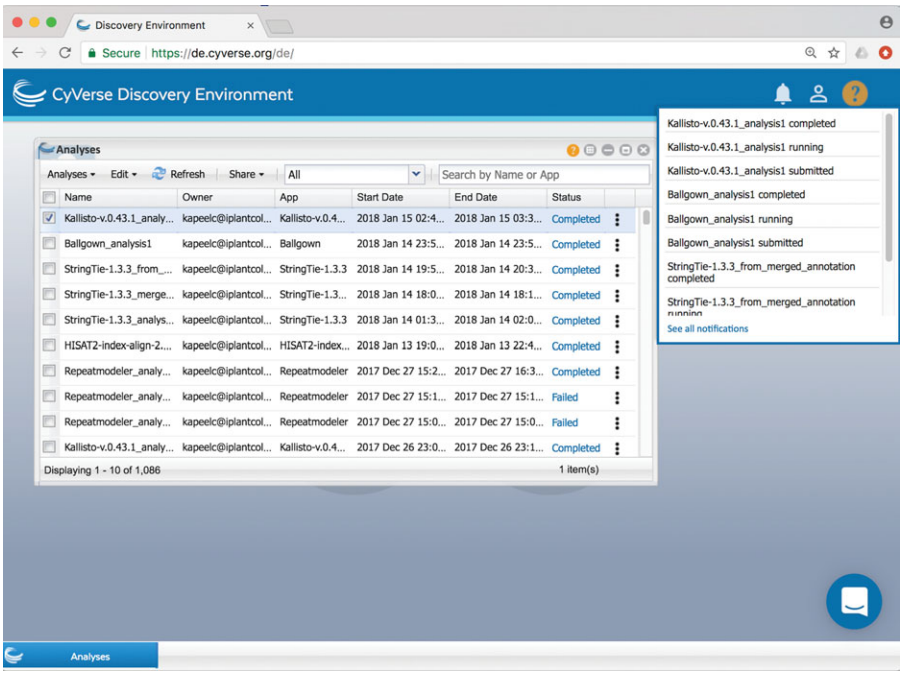

Figure 34 Job status window on CyVerse Discovery Environment. Once the job runs to completion, it is also possible to check the total run time in this window.

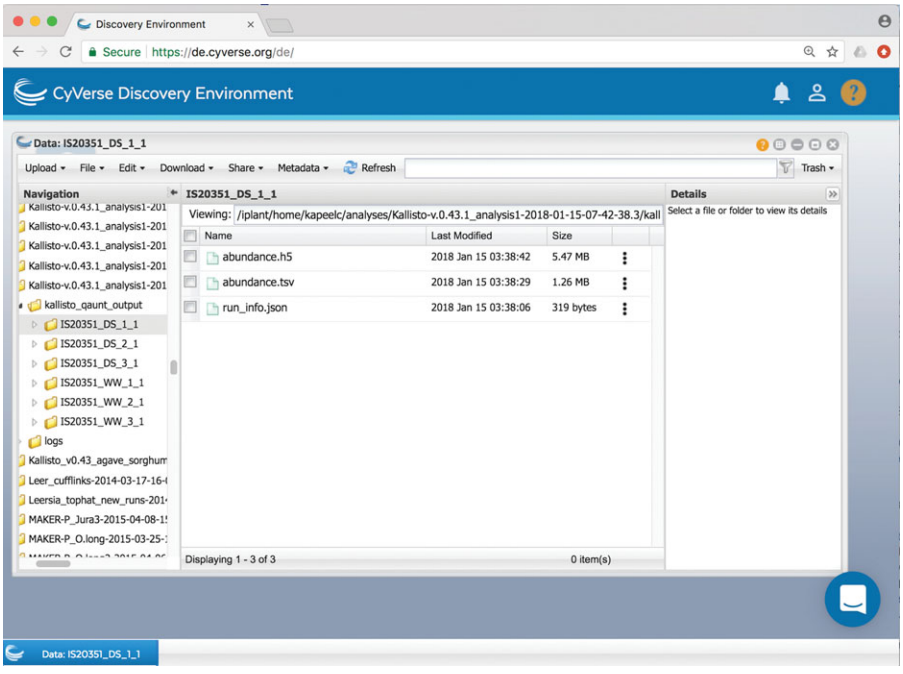

Figure 35 Output folder from `Kallisto-v.0.43.1` runs showing abundance estimates files for each sample.

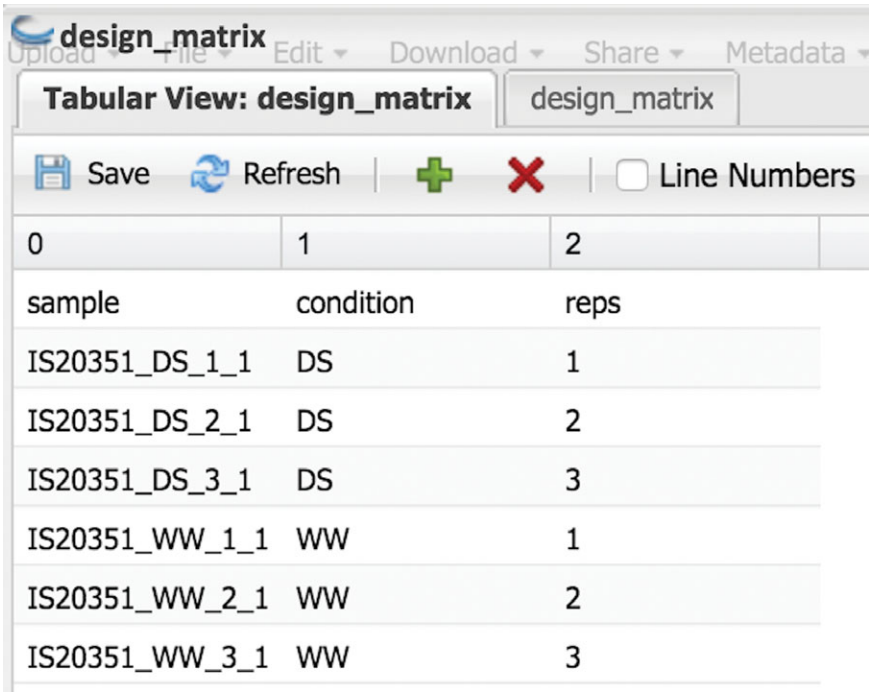

Figure 36 Design matrix file describing the experimental conditions on which differential expression analysis will be performed by Sleuth app.

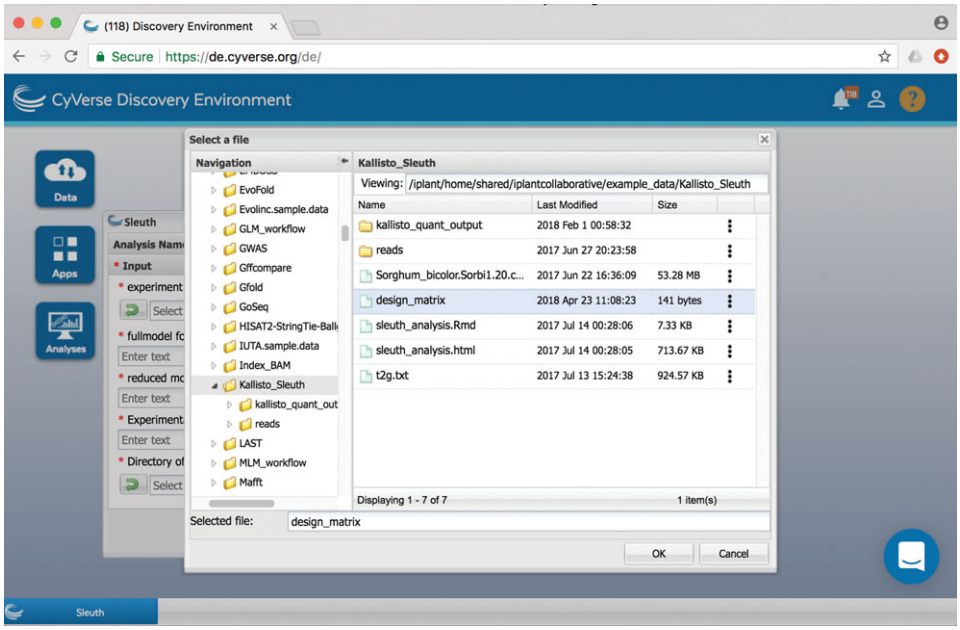

Figure 37 Uploading the design matrix file.

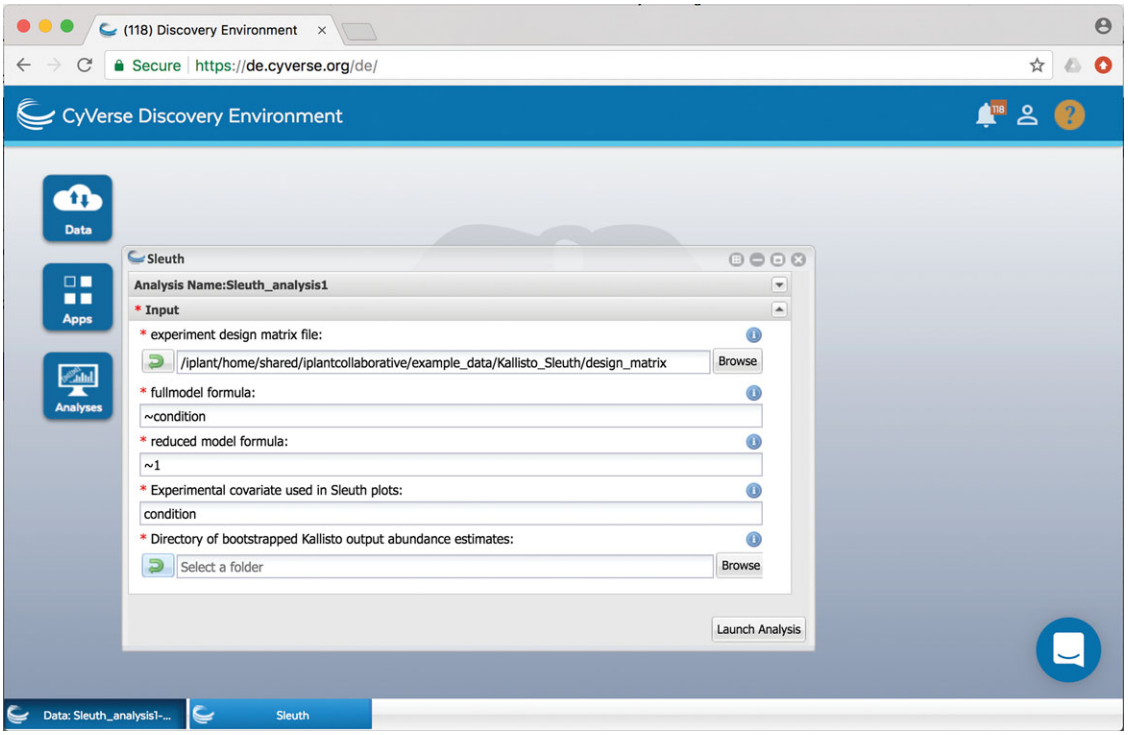

Figure 38 Input options for Sleuth app. Provide formulae for full and reduced model. Make sure to add prefix '~'. Also, provide experimental covariate for plotting.

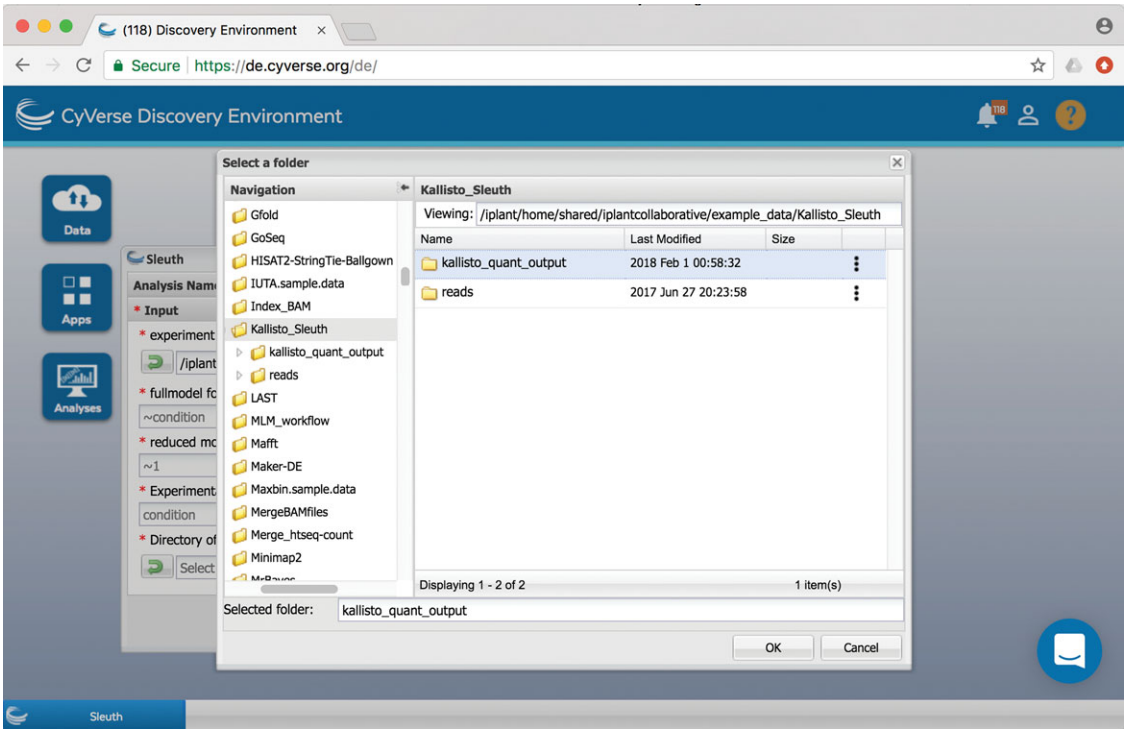

Figure 39 Kallisto output folder with bootstrap transcript abundance estimate for each sample.

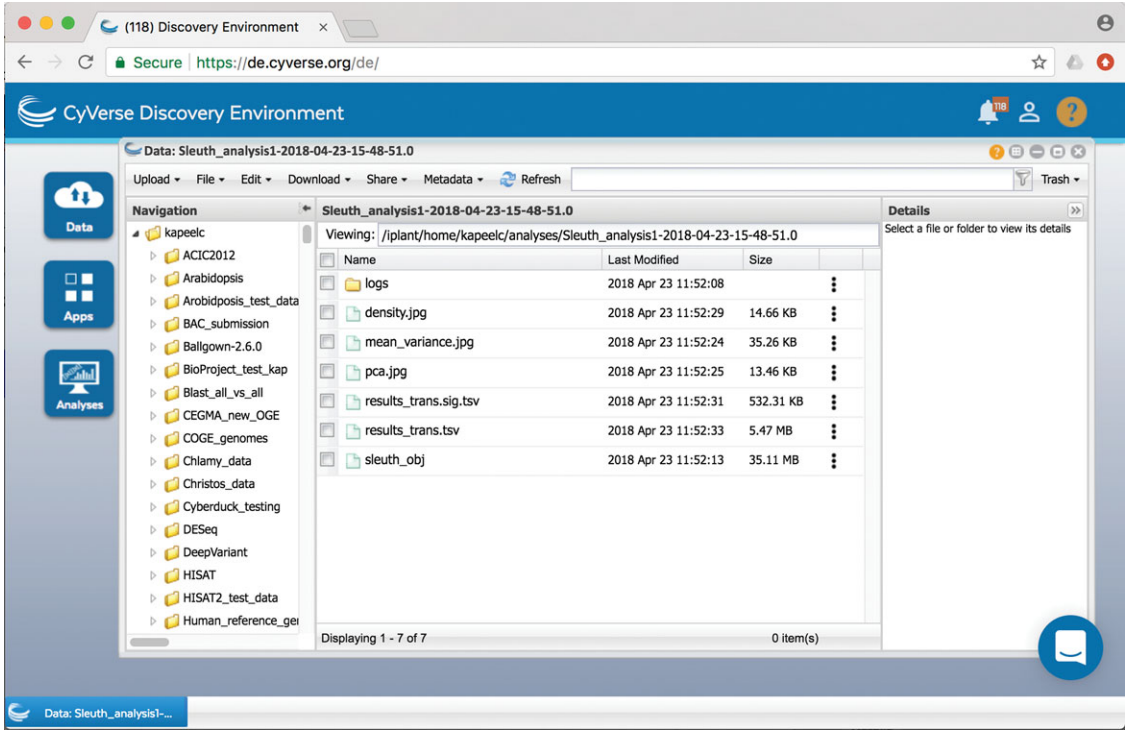

Figure 40 Sleuth output.

13 to 18). To obtain meaningful and reliable results using this protocol, it is recommended that the user have at least three experimental replicates for each sample. Basic Protocol 2 identifies only those reference transcripts that are differentially expressed and where no novel transcripts are predicted.

COMMENTARY

Background Information

Many Web-based analysis tools are run on small-scale computing clusters, resulting in slower run times for submitted jobs. Consequently, they often lack the speed and scaling when handling large datasets. Also, most open-source RNA-seq tools are available from the command line, and therefore require installation of the tools (along with their dependencies) from source or binary on the computing platform. This requires basic to intermediate UNIX skills. Making the workflow available on CyVerse using Docker containers enables easy integration of tools and dependencies. In addition, the easy-to-use Graphical User Interface for each tool on this platform accommodates users of all levels of experience. The two basic protocols described above utilize publicly available tools for RNA-seq data analysis. Each protocol is put into a workflow in which users can use their raw data from multiple samples as inputs, and receive output files without need for file conversion or transfers. These workflows provide customizations for each app, allowing experienced users to build their own customized GUI apps that take advantage of all possible parameters provided by the tool. Also, most downstream differential expression R packages offer rich graphics for visualization of results. The general purpose of these two protocols is to provide users with a more user-friendly workflow in which they can use both graphical user interface and the high-performance computing power provided by the CyVerse Cyberinfrastructure.

Critical Parameters

While many parameters and additional functions of each tool in Basic Protocol 1 and 2 are not fully exploited, the Discovery Environment does permit users to utilize the most common settings for individual software programs in order to run complex pipelines in a simple manner. In Basic Protocol 1, to generate the alignment using HISAT2, it is useful to provide the strand specificity of the RNA-seq reads (if

this information is available), as it helps the aligner to map reads onto the correct strand. Similarly, it is often necessary to remove or trim reads with poor base quality using the trimming parameters available in the app, and select options to report the output alignments to be compatible with downstream assemblers including StringTie or Cufflinks. When using a reference annotation file in transcript assembly with StringTie, a correctly formatted GTF/GFF3 file must be provided; this file can be validated using the sequence ontology based GFF3 specification. For differential expression analysis with Ballgown, when providing the design matrix file, it is critical to have the sample names similar to the folder names of samples in `ballgown_input_files` directory created from the StringTie app. The design matrix files provided should be in tab-delimited text format. The name of the experimental covariate on which Ballgown will perform differential expression analysis should match the column name of samples in the design matrix file.

In Basic Protocol 2, Kallisto requires the user to provide a transcriptome file for indexing, instead of a genome sequence file, and to provide a bootstrap value for better estimates of transcript abundances. When using single-end reads with Kallisto, it is necessary to provide estimated average fragment length and estimated standard deviation of fragment length. This information is usually available from the sequencing center where the library was prepared. For differential expression analysis with Sleuth, when providing the design matrix file it is critical to have the sample names similar to the folder names of samples in the `kallisto_quant_output` directory created by the Kallisto app. The design matrix files provided should be in tab-delimited text format, and the sample column name should be 'sample'. The formulae provided for the full and reduced model should be prefixed by '~'. The name of the experimental covariate on which Sleuth will perform differential expression analysis should match the column name of samples in the design matrix file.

There are also some useful graphics that are generated by both the Ballgown and Sleuth R packages. For additional visualization of results, they can both be run from the command line using Rstudio. Sleuth offers interactive visualization using an *R Shiny* package. Finally, Kallisto is distributed under the BSD 2-clause 'Simplified' License. Please read the license details carefully (*https://pachterlab.github.io/kallisto/download*).

Troubleshooting

It can be confusing to figure out why an analysis failed or does not complete, because there are any number of reasons why this might happen. It could be that one of your input files is corrupted, was not fully uploaded to the DE, or is in the wrong format. Perhaps the problem is with the analysis name, or maybe there is a technical problem with the binary tool used for the app or with the Discovery Environment itself. The URL *https://wiki.cyverse.org/wiki/display/DEmanual/Troubleshooting+an+Analysis* gives some tips on how to troubleshoot the problem so that you can figure out whether it is a problem with your file or with the Discovery Environment, and then help you either work on getting the problem fixed or resubmit another analysis. Another option if the test fails resides in the Discovery Environment analysis window, where the user should click on the 'running/failed/completed' link under status. This will lead the user through several troubleshooting steps. If that does not answer your question, click 'I still need help', and at this point you will be asked to share your analysis. Once you have shared your analysis, the CyVerse support team will attempt to diagnose the problem.

Although the DE Web interface offers multiple ways to upload files for your analysis, it only allows files of maximum size of 1.9 GB per file. For bulk upload to the data store using third-party tools like CyberDuck and iRODS icommands, detailed instructions are available at *https://pods.iplantcollaborative.org/wiki/display/DS/Using+Cyberduck+for+Uploading+and+Downloading+to+the+Data+Store*.

Time Considerations

Basic Protocol 1: Using paired-end reads from *Sorghum* samples with data size of ~2.5 GB and 24 million 100-bp reads for each sample, with a total file size for all samples ~24 GB, the HISAT2 app performs both indexing of the genome (628 MB) and alignment of reads against the reference, which takes ~3.5 hr; transcript assembly for all samples with StringTie app takes ~30 min; merging of transcript assemblies with the reference annotation takes ~3 min; and, finally, differential-expression analysis with Ballgown takes ~4 min to complete. The newer Tuxedo protocol shows an improvement in run time when using the Discovery environment resource to analyze the same test data. By contrast, the older Tuxedo protocol had

longer runtimes: TopHat2, ~13 hr; Cufflinks2, ~5 hr; Cufflinks2-merge, ~5 min; and Cuffdiff2, ~2 hr.

Basic Protocol 2: Using the same data sets, the Kallisto app in the Discovery environment takes ~57 min to index the reference transcriptome (65 Mb) and pseudoalign the reads. Depending on resource availability, launching of an instance on the Atmosphere cloud service takes ~10 min to run Sleuth on the active instance.

Acknowledgments

This work is supported by National Science Foundation under grant DBI-1265383 (for the CyVerse project).

Literature Cited

Bray, N. L., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, *34*(5), 525–527. doi: 10.1038/nbt.3519.

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., ... Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology*, *17*(1), 13. doi: 10.1186/s13059-016-0881-8.

Devisetty, U. K., Kennedy, K., Sarando, P., Merchant, N., & Lyons, E. (2016). Bringing your tools to CyVerse Discovery Environment using Docker. *F1000Res*, *5*, 1442. doi: 10.12688/f1000research.8935.1.

Fracasso, A., Trindade, L. M., & Amaducci, S. (2016). Drought stress tolerance strategies revealed by RNA-Seq in two sorghum genotypes with contrasting WUE. *BMC Plant Biology [Electronic Resource]*, *16*(1), 115. doi: 10.1186/s12870-016-0800-x.

Frazee, A. C., Pertea, G., Jaffe, A. E., Langmead, B., Salzberg, S. L., & Leek, J. T. (2015). Ballgown bridges the gap between transcriptome assembly and expression analysis. *Nature Biotechnology*, *33*(3), 243–246. doi: 10.1038/nbt.3172.

Goff, S. A., Vaughn, M., McKay, S., Lyons, E., Stapleton, A. E., Gessler, D., ... Stanzione, D. (2011). The iPlant Collaborative: Cyberinfrastructure for Plant Biology. *Frontiers in Plant Science*, *2*, 34. doi: 10.3389/fpls.2011.00034.

Kim, D., Langmead, B., & Salzberg, S. L. (2015). HISAT: A fast spliced aligner with low memory requirements. *Nature Methods*, *12*(4), 357–360. doi: 10.1038/nmeth.3317.

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, *9*(4), 357–359. doi: 10.1038/nmeth.1923.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., ... Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. doi: 10.1093/bioinformatics/btp352.

Niedringhaus, T. P., Milanova, D., Kerby, M. B., Snyder, M. P., & Barron, A. E. (2011). Landscape of next-generation sequencing technologies. *Analytical Chemistry*, *83*(12), 4327–4341. doi: 10.1021/ac2010857.

Oliver, S. L., Lenards, A. J., Barthelson, R. A., Merchant, N., & McKay, S. J. (2013). Using the iPlant collaborative discovery environment. *Current Protocols in Bioinformatics*, *42*, 1.22.1–1.22.26. doi: 10.1002/0471250953.bi0122s42.

Pertea, M., Kim, D., Pertea, G. M., Leek, J. T., & Salzberg, S. L. (2016). Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. *Nature Protocols*, *11*(9), 1650–1667. doi: 10.1038/nprot.2016.095.

Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T. C., Mendell, J. T., & Salzberg, S. L. (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*, *33*(3), 290–295. doi: 10.1038/nbt.3122.

Pimentel, H., Bray, N. L., Puente, S., Melsted, P., & Pachter, L. (2017). Differential analysis of RNA-seq incorporating quantification uncertainty. *Nature Methods*, *14*(7), 687–690. doi: 10.1038/nmeth.4324.

Sahraeian, S. M. E., Mohiyuddin, M., Sebra, R., Tilgner, H., Afshar, P. T., Au, K. F., ... Lam, H. Y. K. (2017). Gaining comprehensive biological insight into the transcriptome by performing a broad-spectrum RNA-seq analysis. *Nature Communications*, *8*(1), 59. doi: 10.1038/s41467-017-00050-4.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., ... Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, *7*(3), 562–578. doi: 10.1038/nprot.2012.016.

Yang, I. S., & Kim, S. (2015). Analysis of whole transcriptome sequencing data: Workflow and software. *Genomics & Informatics*, *13*(4), 119–125. doi: 10.5808/gi.2015.13.4.119.

Internet Resources

https://www.youtube.com/watch?v=2TI8huVhMQk&t=2326s

YouTube Webinar describing the theory and Basic Protocols 1 and 2.

https://www.cyverse.org

CyVerse provides services aimed at enabling scientific discovery.

https://de.cyverse.org/de/

Discovery Environment, one of the services provided by CyVerse, allows users to both manage and analyze their data using a graphical user interface.

https://atmo.cyverse.org/application/images

CyVerse Atmosphere Cloud, one of the services provided by CyVerse, allows users to launch their own isolated virtual machine (VM) image and software using compute resources such as CyVerse-provided software suites, preconfigured frequently used analysis routines, relevant algorithms, and datasets.

https://www.cyverse.org/data-store
CyVerse provides services using iRODS technology to securely store and manage data for analysis using CyVerse computational platforms.

https://wiki.cyverse.org/wiki/display/DEmanual/Using+the+Discovery+Environment
Basic tutorial page for using the Discovery Environment.

https://wiki.cyverse.org/wiki/display/atmman/Atmosphere+Manual+Table+of+Contents
Basic tutorial page for using Atmosphere.

https://en.wikipedia.org/wiki/FASTQ_format#Quality
This Web site provides information on FASTQ format and data quality.

https://haroldpimentel.wordpress.com/2014/05/08/what-the-fpkm-a-review-rna-seq-expression-units/
Explanation of expression units used in RNA-seq.

https://github.com/Kapeel/HISAT2/tree/master/v2.1

https://github.com/Kapeel/StringTie

https://github.com/Kapeel/Ballgown

https://github.com/Kapeel/Ballgown_Shinyapp

https://rpubs.com/kapeelc12/Ballgown

https://github.com/Kapeel/sleuth
These repositories provide source code, wrapper scripts, and Docker files for tools described in Basic Protocol 1.

https://github.com/Kapeel/kallisto

https://rpubs.com/kapeelc12/Sleuth

https://github.com/pachterlab/sleuth
These repositories provide source code, wrapper scripts, and Docker files for tools described in Basic Protocol 2.