TCW: Transcriptome Computational Workbench

Carol Soderlund[1]*, William Nelson[1], Mark Willer[1], David R. Gang[2]

[1] BIO5 Institute, University of Arizona, Tucson, Arizona, United States of America, [2] Washington State University, Pullman, Washington, United States of America

Abstract

Background: The analysis of transcriptome data involves many steps and various programs, along with organization of large amounts of data and results. Without a methodical approach for storage, analysis and query, the resulting ad hoc analysis can lead to human error, loss of data and results, inefficient use of time, and lack of verifiability, repeatability, and extensibility.

Methodology: The Transcriptome Computational Workbench (TCW) provides Java graphical interfaces for methodical analysis for both single and comparative transcriptome data without the use of a reference genome (e.g. for non-model organisms). The singleTCW interface steps the user through importing transcript sequences (e.g. Illumina) or assembling long sequences (e.g. Sanger, 454, transcripts), annotating the sequences, and performing differential expression analysis using published statistical programs in R. The data, metadata, and results are stored in a MySQL database. The multiTCW interface builds a comparison database by importing sequence and annotation from one or more single TCW databases, executes the ESTscan program to translate the sequences into proteins, and then incorporates one or more clusterings, where the clustering options are to execute the orthoMCL program, compute transitive closure, or import clusters. Both singleTCW and multiTCW allow extensive query and display of the results, where singleTCW displays the alignment of annotation hits to transcript sequences, and multiTCW displays multiple transcript alignments with MUSCLE or pairwise alignments. The query programs can be executed on the desktop for fastest analysis, or from the web for sharing the results.

Conclusion: It is now affordable to buy a multi-processor machine, and easy to install Java and MySQL. By simply downloading the TCW, the user can interactively analyze, query and view their data. The TCW allows in-depth data mining of the results, which can lead to a better understanding of the transcriptome. TCW is freely available from www.agcol.arizona.edu/software/tcw.

Citation: Soderlund C, Nelson W, Willer M, Gang DR (2013) TCW: Transcriptome Computational Workbench. PLoS ONE 8(7): e69401. doi:10.1371/journal.pone.0069401

Editor: Hector Candela, Universidad Miguel Hernández de Elche, Spain

Received April 15, 2013; **Accepted** June 7, 2013; **Published** July 17, 2013

Funding: This work was supported by National Science Foundation grant IOS-1044821 (http://www.nsf.gov/awardsearch/showAward?AWD_ID = 1044821). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: cari@agcol.arizona.edu

Introduction

With next generation sequencing (NGS), the amount of transcriptome data is increasing rapidly. Typical analyses performed on transcripts are GC-content, open reading frames (ORF), single-nucleotide polymorphisms (SNP), comparisons with protein databases, gene ontology (GO) [1], differential expression (DE), and homology (paralogs and/or orthologs) clustering. Often, publications for transcriptome analysis reference many different programs and perform computations on web sites, which indicates the authors needed to merge the results from the various locations and programs. Most publications do not state what software they used for merging the results, which indicates either that they did not properly reference the software, or they wrote their own scripts and/or used Excel spreadsheets. This causes an 'ad hoc' style of analysis that can lead to human error, loss of data and results, inefficient use of time, and lack of verifiability, repeatability and extensibility. Moreover, this approach does not make the data and results easily available on the web in a queryable form for the community.

The Transcriptome Computational Workbench (TCW) aims to make analysis more systematic by consolidating data, analysis and results. Towards this end, TCW contains two manager programs: singleTCW has a graphical interface for building a database of annotated sequences with DE results for a single species, and multiTCW has a graphical interface for building a database of multiple species with comparison results. It uses external programs when appropriate, where most are packaged within the TCW for ease of installation. Except for downloading annotation databases from the web (for which scripts are supplied), the TCW is a web-free program so the user is not dependent on having a good Internet connection or contention with other web users. For large projects, it is beneficial to have a high-end computer, but a 32 CPU 2.4 ghz AMD machine with 128G of RAM and 7TB internal disk space can now be purchased for less than $6000. It is helpful to have a system administrator to configure the machine, but it is now common for biology departments to have such support. To keep installation simple, the TCW uses the common platforms Java, MySQL, and R (optional for differential expression analysis).

Many transcriptome publications show 'big picture' results, e.g. a chart of the major GO categories. Though this information is worthwhile, it is equally important to be able to drill down into the data and view exactly what the alignments look like, all the annotation hits (not just the best e-value), etc. Interactive data mining can provide a better understanding of the transcriptome and lead to new and better experiments. The TCW provides this interactive exploratory environment within both singleTCW and multiTCW. A big advantage of using Java is that the project analysis can be run locally for speed, but then results can be made publically available as an applet on the web (albeit, startup time is slower).

The TCW takes as input Sanger reads, 454 reads and/or pre-assembled transcripts (e.g. Illumina) with read counts. It does not perform assembly of high volume short reads as there are many good software programs to perform that task along with computing the read counts (see [2] for a good review). Hence, the TCW can assemble a mix of long reads and transcripts, computing the counts of the reads and integrating the read counts of the transcripts. It will also work directly with a pre-assembled transcript set (i.e. no assembly needed). Though this manuscript is written for transcriptome analysis, the system can also be used for the annotation and DE analysis of shotgun proteomics, i.e. in both cases there are sequences and abundance levels. This would be especially advantageous when joint transcriptome and proteome experiments are performed, as the analysis would be the same for both datasets. To our knowledge, there is no other program that can work with this set of inputs.

The TCW is a combination of pipeline software with a graphical interface and interactive graphics of the results for both single and multiple species. To our knowledge, there is no other software that performs both single and multiple analyses with interactive display. However, there are freely available software programs that have overlapping functionality. As the TCW is specifically written for "project-specific desktop interactive transcriptome analysis", we compare its features with similar freely available downloadable interactive programs (i.e. not web-based or configurable pipelines).

Both TCW and Blast2GO [3] provide annotation from protein databases and GO assignments along with interactive graphics of the results, but they are otherwise quite different. Using Blast2GO, the user does not need a multiprocessor machine to run the blasts as it uses a cloud server. They do not need to download databases, and the results include InterPro [4] and pathway data. Using TCW, the user can easily configure it to use their annotation databases of choice (protein or nucleotide), it results in an in-house MySQL database of all data and results with differential expression analysis and comparisons across transcriptomes, and it has the ability to provide the results on the web. Both TCW and Blast2GO are written in Java and aim to provide an easy interface for the user. A scenario that would utilize the best of both applications is to use the TCW to output to a file the sequences from the differentially expressed annotated transcripts, and then further analyze the sequences with Blast2GO for pathway, InterPro or other Blast2GO specific features.

An important set of R statistical packages for transcriptomics relates to normalization and DE analysis. The RobiNA software [5] has packaged the edgeR [6], DESeq [7], and EDASeq [8] commands into a Java program, along with executing Bowtie [9] to align the short reads, but does not provide other features of the TCW. The TCW starts processing after the alignment but before normalization and DE analysis. It has packaged the edgeR, DESeq, EDASeq, and DEGseq [10] commands along with the GOseq [11] commands for GO differential expression analysis.

There are a number of programs for the computation of orthologs (e.g. orthoMCL [12] and InParanoid [13]), but they do not have a desktop interactive display. The one exception is OrthoInspector [14], which is a Java package with both a novel algorithm and graphics. In contrast, the TCW does not have its own clustering algorithm except for a simple transitive closure based on BLAST e-values, which provides a way to view the most similar sequences regardless of inparalogs, outparalogs or orthologs. The aim of TCW is more general purpose, as it takes as input any type of clusters (homologous, co-expression, etc.) for combined query and display. OrthoMCL is one of the most widely used ortholog clustering programs, which works for multiple species input. However, it does not provide a way to view the results and assumes knowledge of the Unix command-line environment as it has many steps to perform. Therefore, the TCW has packaged this program for orthologous clustering to make it easier for the user to compute and display orthoMCL clusters.

The TCW provides query and display of clusters, but not the highly graphical displays of CytoScape and its plugins. It provides query and display of read counts (raw or RPKM) along with DE values, but it does not provide the extensive analysis and display of programs such as MeV [15]. Instead, the TCW provides various forms of output that can be used as input into these Java desktop programs in order that a biology laboratory can create an "environment" of interactive graphical software to rigorously explore their data and results.

In genomics, there is a range of software for web-based computing where some sites provide full service (i.e. hardware, software, storage, backups), such as Galaxy [16] and iPlant [17], or just the hardware and software such as ArrayExpressHTS [18] and MyRNA [19]. This is part of the wider trend towards 'cloud computing' or 'software as a service'. The advantages of web-based computing are that the lab does not need to purchase a high-end machine, install software, or (in some cases) provide backups and storage. The advantages of local computing on a high-end machine (preferably that belongs to the project) are that all data is in one place, the lab can more easily collaborate with computational scientists to experiment with new software approaches, and interactive local programs (e.g. written in Java) can be much faster for exploring the data. Local computation also reduces privacy concerns and complications related to data transfer, task-sharing, and detection and re-running of failed processes; moreover, with the steadily-falling price of standalone hardware, the cost is not necessarily higher than a cloud solution. As further discussed in the Discussion, we contend that genomics needs a mix of cloud based and local computing for years to come to give biologists maximal flexibility for exploratory research.

Much of the TCW software has grown out of years of collaborating with biologists on transcriptome analysis, where the original program was referred to as PAVE (Program for Assembling and Viewing ESTS) [20]. Due to its evolution to next-gen sequencing support and comparative analysis, the name was changed to Transcriptome Computational Workbench. This upgraded software was developed for a plant-based project (www.plantrhizome.org), which is sequencing the transcriptomes of rhizomes and other tissues from over a dozen plants to determine what genes distinguish the rhizome from other tissues. None of these plants have sequenced genomes, and only a few have closely related sequenced genomes. Due to the amount of data being generated, it was important to develop a methodical and easy system to use, hence, the TCW. Results are published elsewhere (e.g. He et al. [21]), but timing results will be included here from the rhizome project.

Methods

TCW is composed of five graphical interfaces: runSingleTCW for building the singleTCW database, runDE for adding DE results to the database, viewSingleTCW for query and display of the results, runMultiTCW for building a comparison database, and viewMultiTCW for query and display of the results.

Build Single TCW Database (runSingleTCW)

Input. The only required input is one or more FASTA files of sequences. This program works for any of the following scenarios: (1) assemble Sanger and/or 454 reads, which generates consensus sequences and read counts; (2) import transcripts (e.g. assembled Illumina consensus sequences or gene models) with read counts, which may have replicas for DE analysis; (3) assemble multiple transcript libraries, maintaining the read counts; (4) assemble Sanger and/or 454 reads with transcripts, maintaining the transcript read counts and computing the Sanger and/or 454 read counts; and (5) import protein sequences with abundances from proteomic experiments. We have used the TCW for all five of these scenarios. The manager provides the following steps (see Figure 1):

Step 1. Load libraries. The user defines one or more read or transcript files, with optional quality files. For transcripts, they may also define the read count files (multiple files or one delimited file). Replicas (if any) are grouped per library, with values summed for display and used separately for DE analysis. The user defines a short column heading for each library to be used in viewSingleTCW. The user also enters the metadata title, species, cultivar, strain, tissue, treatment, sequencing lab, and year to be displayed on the viewSingleTCW overview page.

Step 2. Assemble or instantiate. As detailed in Soderlund et al. [20], the assembly algorithm performs multiple iterations of self-blast and assembly using CAP3 [22]. It was initially developed for Sanger sequences, where mate pairs are retained in the same contigs. It was then extended for 454 reads, which tend to have a higher abundance of reads compared to Sanger, so the algorithm removes reads contained in another from assembly, and after assembly, they are added back into contigs as 'buried' reads. Since the original publication, the code has been ported from Perl to Java, plus the following three features have been added. First, SNPs are computed using a binomial p-value based on the number of confirming reads, the depth of the contig at that base, and the estimated base call error rate. Second, the code was extended for already-assembled reads (e.g. Illumina) by allowing transcripts with their optional read counts to be instantiated (i.e. no assembly is performed) or an assembly may be performed while maintaining input read counts. Third, normalization and differential expression statistics are computed, as discussed below.

Step 3. Annotation. The user can define one or more protein or nucleotide databases to blast against, and attach a 'taxonomy' identifier to each to be used as a short tag for query on a specific database in viewSingleTCW. If the UniProt [23] taxonomic databases are used, each entry is given a tag to identify the taxonomy (i.e. plant, invertebrate, etc.) and whether is it SwissProt or TrEMBL. Additionally, the GO terms, KEGG [24], EC [25], and Pfam [26] identifiers are extracted from the UniProt '.dat' files and added to the TCW database for query and display. The GO analysis uses the GO tree downloaded from www.geneontology.org to extract the levels and descriptions.

During annotation, the GC content and ORFs are computed. The user also has the option to have a self-blast of the sequences run and all similar pairs identified, i.e. this is a good way to estimate how stringent the assembly was.

Normalization and Differential Expression Results (runDE)

Step 2 above automatically computes RPKM (Reads Per Kilobase of transcript per Million mapped reads) normalization on the raw values. As Mortazavi et al. [27] observed, two transcripts may have the same actual expression level but the longer transcript will have more reads than the shorter one, falsely giving it a higher apparent expression level; therefore, the length must be taken into account. Step 2 also computes the Poisson-based R statistic [28], which was designed for Sanger EST data and takes into account library size and frequency of the read count in all libraries. Though this should not replace use of a more rigorous statistic (e.g. as provided by the DE R packages), it has the virtues of being computationally inexpensive and testing a whole group at once, rather than performing pairwise comparisons.

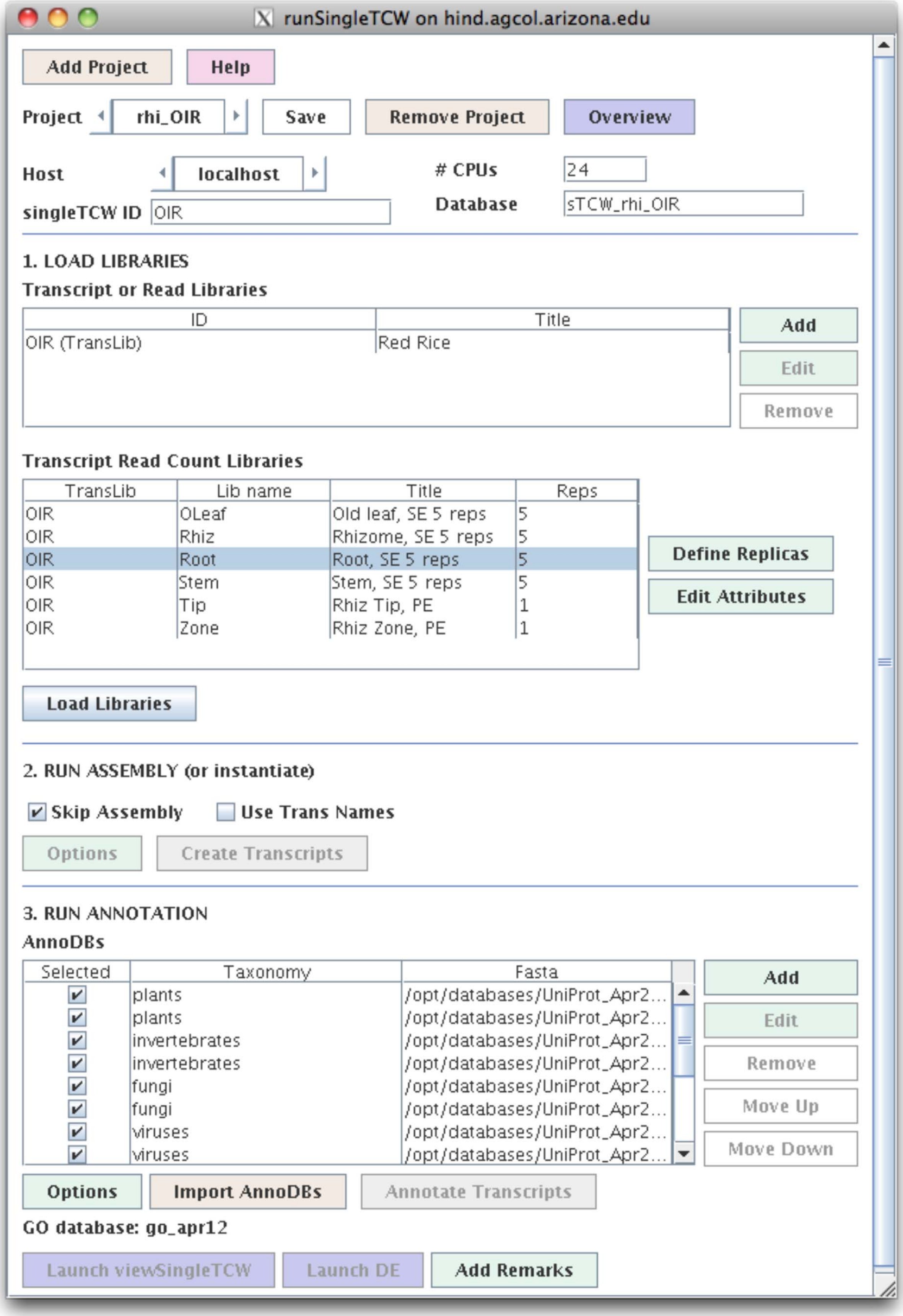
With the advent of next-gen sequencing, a number of normalization and DE analysis programs have been written in R to take into account the unique attributes of RNAseq data. The EDASeq package performs (among other functions) normalization that takes into account GC-content biases because GC-rich and GC-poor reads tend to be under-represented, and GC-content effects tend to be lane-specific [8]. The edgeR method uses variability between biological replicas to compute the dispersion parameter for a negative-binomial model, which improves the reliability of the differential expression statistic between two libraries; though edgeR can be used without replicas, they strongly suggest having biological replicas [6]. DEGseq provides several different methods, of which three are implemented in TCW: Fisher's exact test, likelihood ratio test, and an MA-plot-based method [10]. DESeq [7] also uses the biological replicas to compute the dispersion [29]. GOseq [11] computes DE p-values for GO terms based on enrichment of underlying DE transcripts. There are other R packages for DE analysis (e.g. BaySeq [29]), which are not included within the TCW but we would add upon request, or could be added to the downloadable code.

To run any of these computations requires some knowledge of R and often requires many commands to be performed. To simplify this analysis for the user, the commands are bundled within runSingleTCW. The user can request default or the EDAseq normalization (see Figure 2), i.e. all three packages have default normalization or can take as input normalized values. Some of these programs provide useful graphing and exploratory functions, therefore, the execution finishes with the R session active at the terminal along with the commands to generate selected graphs, e.g. edgeR provides a scatter plot which can be viewed with "plotBCV(d)". Columns can be added and removed with results from the different analyses so the user can visually evaluate the results in viewSingleTCW.

Though the user can select multiple libraries for each of the two groups, the code treats all the sequences in each group as replicas, hence we recommend that the user perform all pairwise comparisons of interest, and then use viewSingleTCW to search for all or any of the p-values that pass the user-supplied cutoff. For example, with the rhizome data, to ask what transcripts are differentially expressed in rhizome tips compared to stem and root, p-values are computed for each pair (tip-root and tip-stem), then the query requests all transcripts that have a p-values less than a user-supplied value for both columns of p-values (results in Figure 3).

Single TCW Query and Display (viewSingleTCW)

As shown in Figure 3, the graphical interface is similar to the BioMart [30] approach, where the user selects the columns (attributes) to view and the filters for searches, where the salient filters are on RPKM values, DE p-values, fold changes, ORFs, and



Project: rhi_OIR

Host: localhost

singleTCW ID: OIR

CPUs: 24

Database: sTCW_rhi_OIR

1. LOAD LIBRARIES

Transcript or Read Libraries

ID	Title
OIR (TransLib)	Red Rice

Transcript Read Count Libraries

TransLib	Lib name	Title	Reps
OIR	OLeaf	Old leaf, SE 5 reps	5
OIR	Rhiz	Rhizome, SE 5 reps	5
OIR	Root	Root, SE 5 reps	5
OIR	Stem	Stem, SE 5 reps	5
OIR	Tip	Rhiz Tip, PE	1
OIR	Zone	Rhiz Zone, PE	1

2. RUN ASSEMBLY (or instantiate)

☑ Skip Assembly ☐ Use Trans Names

3. RUN ANNOTATION

AnnoDBs

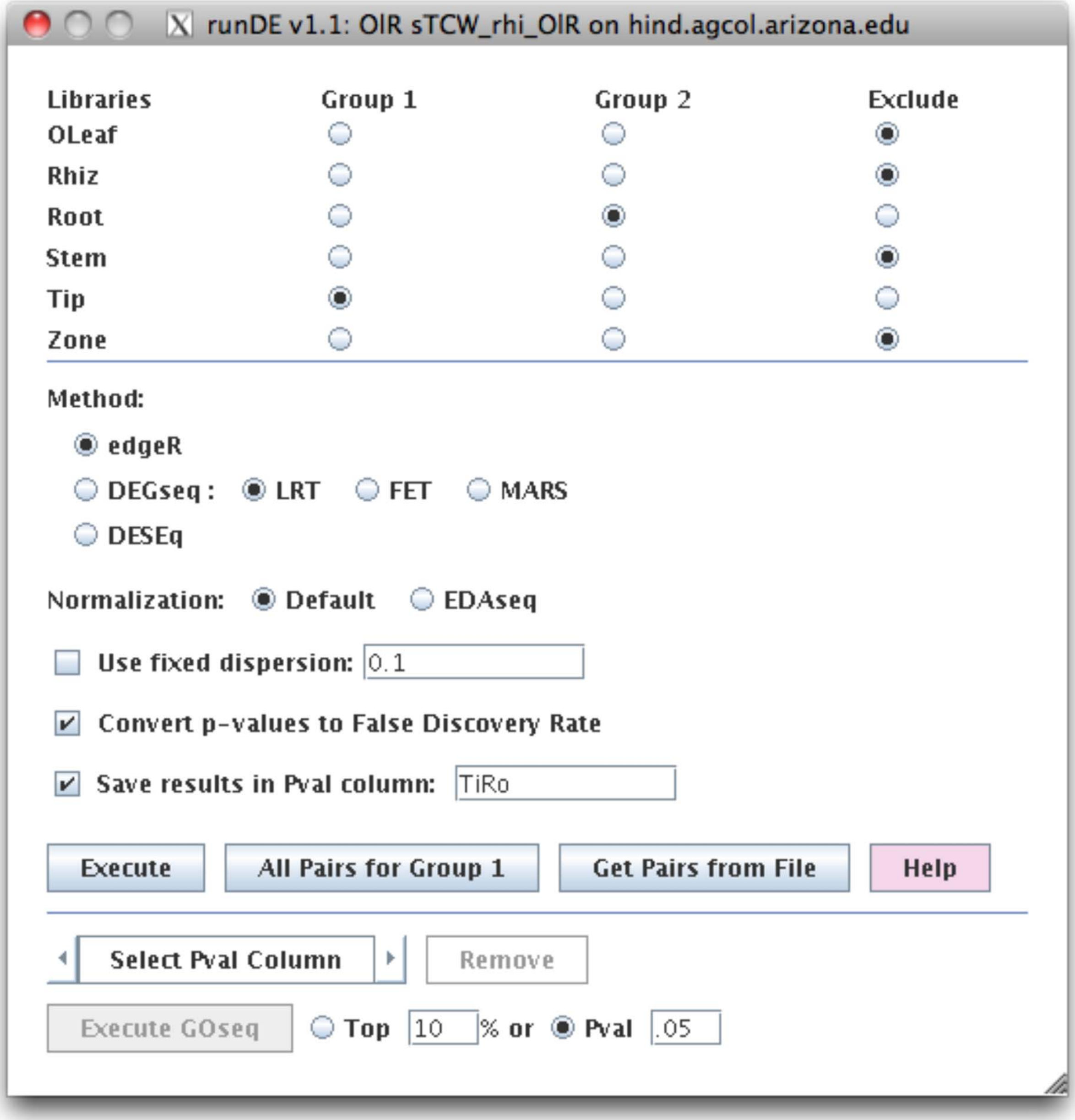
Selected	Taxonomy	Fasta
☑	plants	/opt/databases/UniProt_Apr2…
☑	plants	/opt/databases/UniProt_Apr2…
☑	invertebrates	/opt/databases/UniProt_Apr2…
☑	invertebrates	/opt/databases/UniProt_Apr2…
☑	fungi	/opt/databases/UniProt_Apr2…
☑	fungi	/opt/databases/UniProt_Apr2…
☑	viruses	/opt/databases/UniProt_Apr2…
☑	viruses	/opt/databases/UniProt_Apr2…

GO database: go_apr12

Figure 1. The runSingleTCW interface. This shows the configuration for building the red rice singleTCW database. The ''Add'' button in the LOAD LIBRARIES section opens a window (not shown) for defining the location of the sequence file and read count files; when the window is closed, information about the read count files is automatically entered into the ''Transcript Read Count Libraries'' table. The ''Define Replicas'' button opens a window to allow the user to define the replicas, which results in an updated ''Transcript Read Count Libraries'' table. In this case, there were 6 tissues types where the first 4 have 5 replicas each. As there was just one pre-assembled transcript data set, it was instantiated without assembly. The 13 annotation databases along with the GO database were identified in the third step.
doi:10.1371/journal.pone.0069401.g001

attributes of the sequences such as length. The search results in a table of sequence identifiers and associated information, where the table is ''full-featured'', i.e. columns can be moved and multiple columns sorted.

A row in the query result table can be selected to view details of a sequence, e.g. alignment of reads to the consensus (assembled only), alignment of hits from the annotation databases, listing of hits, the GO tree of the hits, and other information about the sequence such as ORF and GC content. The alignment of the



runDE v1.1: OlR sTCW_rhi_OlR on hind.agcol.arizona.edu

Libraries	Group 1	Group 2	Exclude
OLeaf	○	○	●
Rhiz	○	○	●
Root	○	●	○
Stem	○	○	●
Tip	●	○	○
Zone	○	○	●

Method:

● edgeR

○ DEGseq : ● LRT ○ FET ○ MARS

○ DESEq

Normalization: ● Default ○ EDAseq

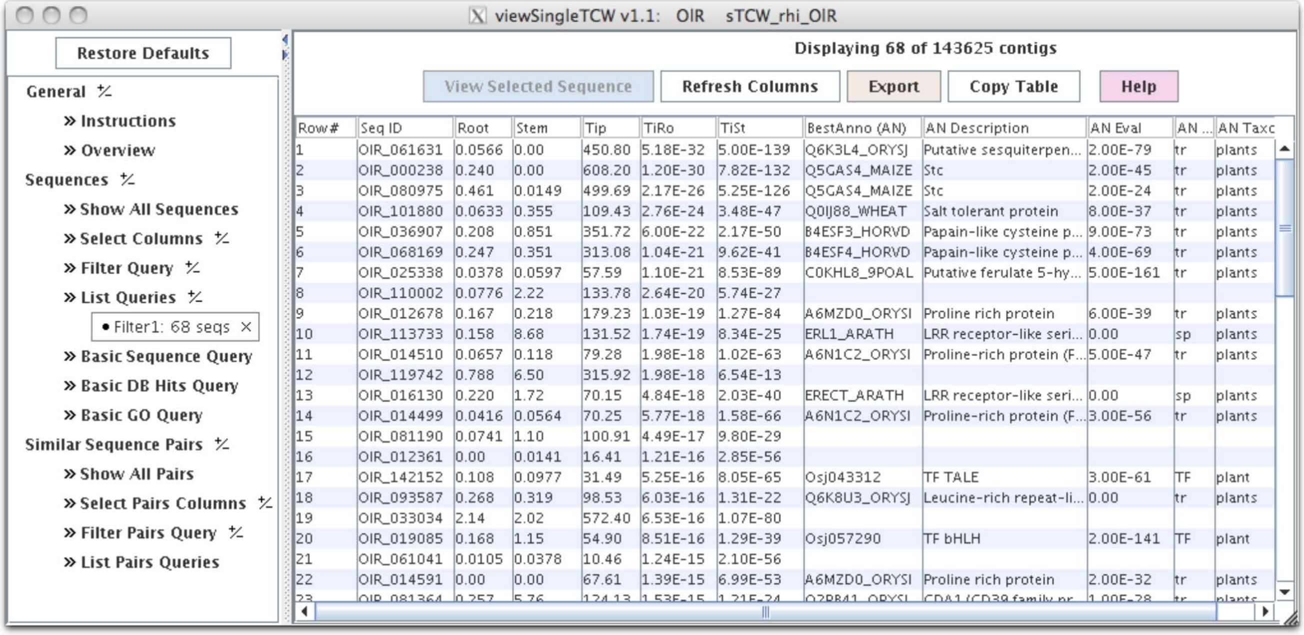
□ Use fixed dispersion: 0.1

☑ Convert p-values to False Discovery Rate

☑ Save results in Pval column: TiRo

○ Top 10 % or ● Pval .05

Figure 2. The runDE interface. This shows the setup to compute the differential expression (DE) between tip and root using the edgeR method with default normalization. The results will be written into a database column called ''TiRo''. Once all DE values are computed, the ''All Pval Columns'' can be selected followed by ''Execute GOSeq'' to compute the corresponding GO differential expression p-values.
doi:10.1371/journal.pone.0069401.g002



viewSingleTCW v1.1: OIR sTCW_rhi_OIR

Displaying 68 of 143625 contigs

Row#	Seq ID	Root	Stem	Tip	TiRo	TiSt	BestAnno (AN)	AN Description	AN Eval	AN ...	AN Taxc
1	OIR_061631	0.0566	0.00	450.80	5.18E-32	5.00E-139	Q6K3L4_ORYSJ	Putative sesquiterpen...	2.00E-79	tr	plants
2	OIR_000238	0.240	0.00	608.20	1.20E-30	7.82E-132	Q5GAS4_MAIZE	Stc	2.00E-45	tr	plants
3	OIR_080975	0.461	0.0149	499.69	2.17E-26	5.25E-126	Q5GAS4_MAIZE	Stc	2.00E-24	tr	plants
4	OIR_101880	0.0633	0.355	109.43	2.76E-24	3.48E-47	Q0IJ88_WHEAT	Salt tolerant protein	8.00E-37	tr	plants
5	OIR_036907	0.208	0.851	351.72	6.00E-22	2.17E-50	B4ESF3_HORVD	Papain-like cysteine p...	9.00E-73	tr	plants
6	OIR_068169	0.247	0.351	313.08	1.04E-21	9.62E-41	B4ESF4_HORVD	Papain-like cysteine p...	4.00E-69	tr	plants
7	OIR_025338	0.0378	0.0597	57.59	1.10E-21	8.53E-89	C0KHL8_9POAL	Putative ferulate 5-hy...	5.00E-161	tr	plants
8	OIR_110002	0.0776	2.22	133.78	2.64E-20	5.74E-27					
9	OIR_012678	0.167	0.218	179.23	1.03E-19	1.27E-84	A6MZD0_ORYSI	Proline rich protein	6.00E-39	tr	plants
10	OIR_113733	0.158	8.68	131.52	1.74E-19	8.34E-25	ERL1_ARATH	LRR receptor-like seri...	0.00	sp	plants
11	OIR_014510	0.0657	0.118	79.28	1.98E-18	1.02E-63	A6N1C2_ORYSI	Proline-rich protein (F...	5.00E-47	tr	plants
12	OIR_119742	0.788	6.50	315.92	1.98E-18	6.54E-13					
13	OIR_016130	0.220	1.72	70.15	4.84E-18	2.03E-40	ERECT_ARATH	LRR receptor-like seri...	0.00	sp	plants
14	OIR_014499	0.0416	0.0564	70.25	5.77E-18	1.58E-66	A6N1C2_ORYSI	Proline-rich protein (F...	3.00E-56	tr	plants
15	OIR_081190	0.0741	1.10	100.91	4.49E-17	9.80E-29					
16	OIR_012361	0.00	0.0141	16.41	1.21E-16	2.85E-56					
17	OIR_142152	0.108	0.0977	31.49	5.25E-16	8.05E-65	Osj043312	TF TALE	3.00E-61	TF	plant
18	OIR_093587	0.268	0.319	98.53	6.03E-16	1.31E-22	Q6K8U3_ORYSJ	Leucine-rich repeat-li...	0.00	tr	plants
19	OIR_033034	2.14	2.02	572.40	6.53E-16	1.07E-80					
20	OIR_019085	0.168	1.15	54.90	8.51E-16	1.29E-39	Osj057290	TF bHLH	2.00E-141	TF	plant
21	OIR_061041	0.0105	0.0378	10.46	1.24E-15	2.10E-56					
22	OIR_014591	0.00	0.00	67.61	1.39E-15	6.99E-53	A6MZD0_ORYSI	Proline rich protein	2.00E-32	tr	plants
23	OIR_081364	0.257	5.76	124.13	1.53E-15	1.21E-24	Q2PB41_ORYSI	CDA1 (CD39 family pr...	1.00E-28	tr	plants

Figure 3. The viewSingleTCW interface. This shows the main table, where the filter was set to show all sequences where tip is differentially expressed compared to root and stem using a 1E-10 cutoff. The columns are: Root, Stem and Tip are the RPKM values; TiRo and TiSt are DE p-values between tip-root and tip-stem, respectively; BestAnno (AN) is the best annotation for the sequence (i.e. not containing phrases such as "uncharacterized protein"); and the last four columns are the BestAnno description, BLAST e-value, the DB type ('tr' = TrEMBL, 'sp' = Swiss-Prot, 'TF' = plantTFDB), and the taxonomic type (all these best hits are from plants).

doi:10.1371/journal.pone.0069401.g003

annotation hits to the sequences uses dynamic programming, and allows the user to see exactly how the hits aligned. The sequence can automatically be sent to the UniProt website to be aligned to all current sequences in that database.

A Basic DB hit query allows search by hit identifier, hit description, or attributes; the results are shown in a table of hit identifiers (i.e. UniProt IDs), with the number of sequences that contained the hit along with the associated UniProt information. The user can drill down by selecting a row and view all the associated sequences. A Basic GO query is very similar; filters include GO number and description keywords, along with DE p-values from GOseq, and the "level:" of the GO (an approximate concept as the GO is not structured as a tree [31]).

Build multiTCW Database (runMultiTCW)

Input. One or more databases built with singleTCW. As shown in Figure 4, there are three steps to building a multiTCW database:

1. Import and extract coding sequence. The user selects one or more singleTCW databases. For each database, the user selects the SMAT file for ESTscan [32] to apply to the sequences (the SMAT file is created from related gene Genbank files using an ESTscan supplied script). ESTscan is executed on all sequences and the resulting proteins loaded into the database. If the singleTCW contains protein sequences, then ESTscan is not run. All annotation hits from the singleTCWs are added to the multiTCW database.

2. Run BLAST and filter. The user can supply a blast file of the self-blast of the combined FASTA file produced by step 1, or they can request that the TCW execute the blast. As some of the sequences from a given database can be almost identical (depending on the approach used to create the sequences), the user has the option of removing very similar sequences (based on user supplied parameters of overlap and similarity).

3. Clustering. The user can select one or more clusterings to be performed. For example, they can request orthoMCL with inflation = 4, orthoMCL with inflation = 1.5, transitive closure with overlap cutoff = 80% and similarity = 80%, and upload a file of clusters generated by some other method. This would result in four cluster sets in the database. The annotation that hits the most sequences in a given cluster is assigned to it. Clusters may be removed or added at any time.

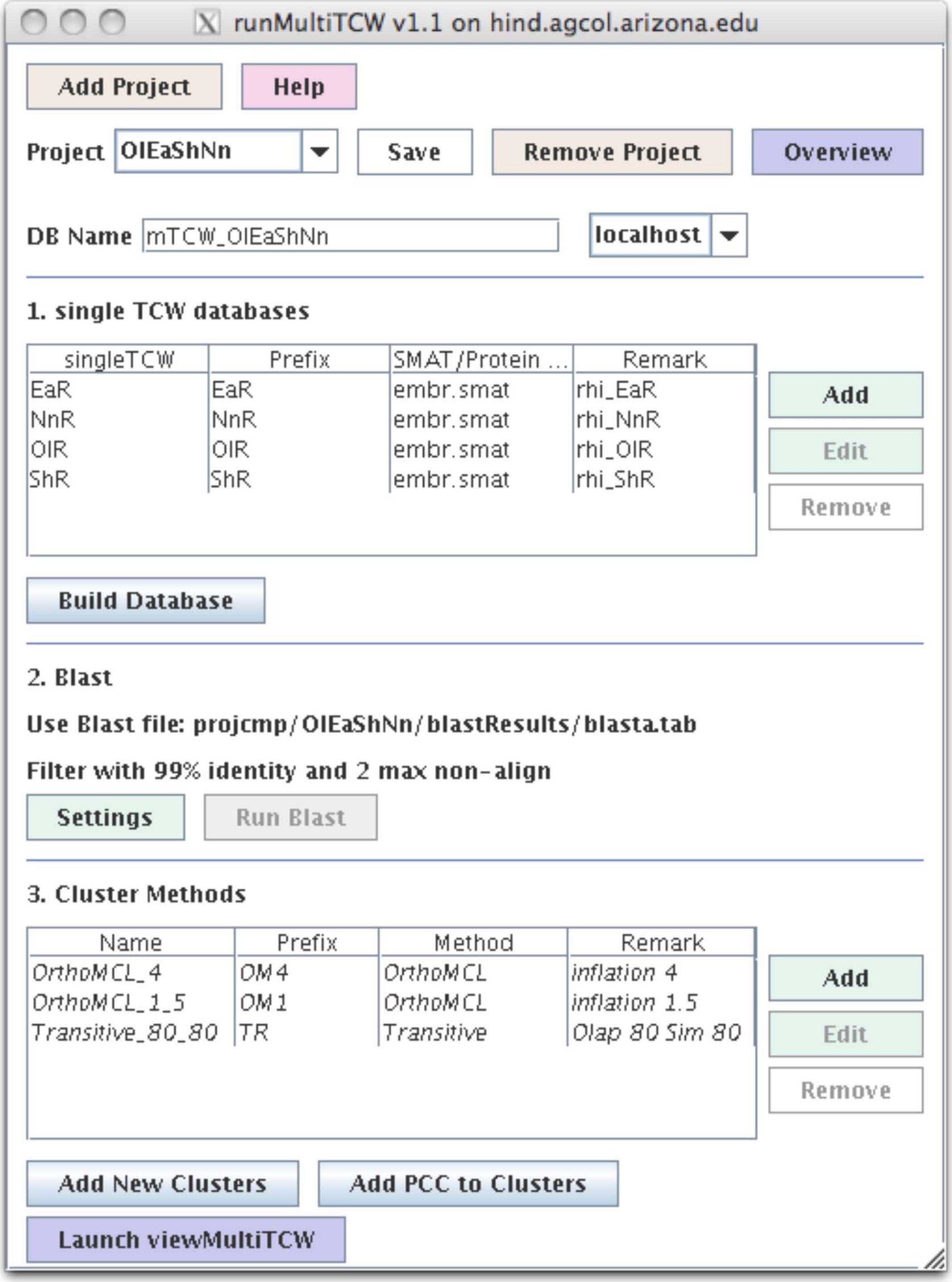
Multi TCW Query and Display (viewMultiTCW)

The filters allow the user to select a cluster set to view (e.g. orthoMCL with inflation 4), select clusters that have a specific species composition (e.g. only clusters with red rice and horsetail sequences), or select based on RPKM composition. Selecting a cluster results in a full-featured table of the sequences contained in the cluster, as shown in Figure 5. The sequences can be aligned with MUSCLE [33] or the sequences can be pairwise aligned. In both cases, the alignments can be viewed in graphical or sequence detail. A sequence can be selected to view more detail, e.g. all of its annotation hits. The columns of the sequence and cluster tables work differently from singleTCW, as they can be interactively added and removed from the table without a new query of the database.

The TCW Package

A major objective of the TCW is ease of use, starting with installation. Towards this end, everything is kept as simple as possible so that the biologist can download the package and immediately try it on the demo.

Platform. The TCW uses the common platforms of Java and MySQL. There are two Perl scripts, one to download all desired



Project OlEaShNn

DB Name mTCW_OlEaShNn localhost

1. single TCW databases

singleTCW	Prefix	SMAT/Protein ...	Remark
EaR	EaR	embr.smat	rhi_EaR
NnR	NnR	embr.smat	rhi_NnR
OlR	OlR	embr.smat	rhi_OlR
ShR	ShR	embr.smat	rhi_ShR

2. Blast

Use Blast file: projcmp/OlEaShNn/blastResults/blasta.tab

Filter with 99% identity and 2 max non-align

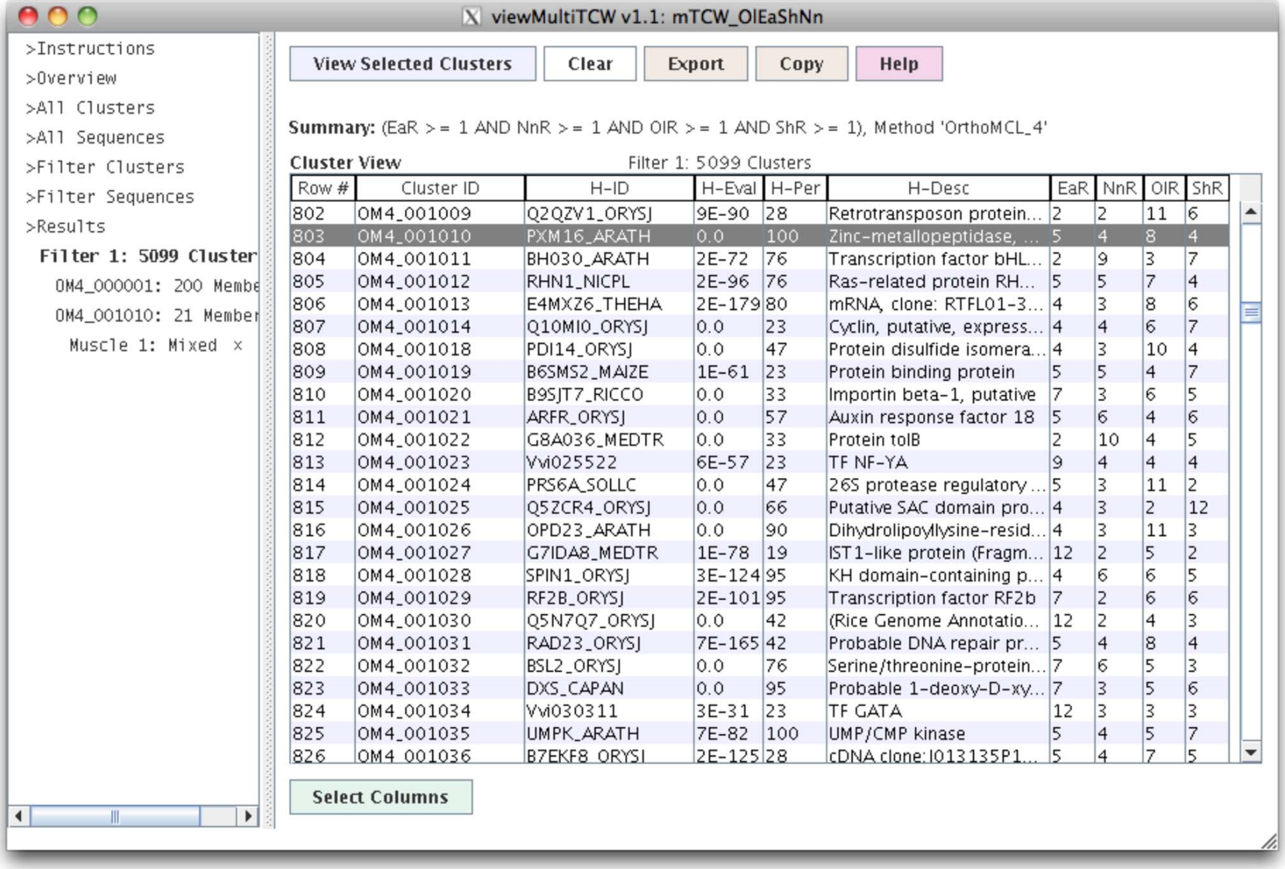
3. Cluster Methods

Name	Prefix	Method	Remark
OrthoMCL_4	*OM4*	*OrthoMCL*	*inflation 4*
OrthoMCL_1_5	*OM1*	*OrthoMCL*	*inflation 1.5*
Transitive_80_80	*TR*	*Transitive*	*Olap 80 Sim 80*

Figure 4. The runMultiTCW interface. This shows the configuration for building a comparison multiTCW database of 4 rhizome singleTCW databases. Three cluster methods were performed: orthoMCL with inflation 1.5, orthoMCL with inflation 4, and transitive with 80% similarity and 80% overlap. The italicized methods indicate they have been added to the database; new clusters can be added and will not be italicized until processed.
doi:10.1371/journal.pone.0069401.g004

UniProt taxonomic databases and extract the sequences and the second to download the GO tree and merge UniProt information with it.

External software packages. TCW uses the external programs Blast, CAP3, MUSCLE, ESTscan and orthoMCL; all external software packages except BLAST are included in the package (TCW can use the old BLAST or new BLAST+). In case of version updates, as long as there is not a format change, the user can easily replace external binaries. For ESTscan, the user needs to make a SMAT matrix from similar sequences (instructions are provided); all other steps are run by runMultiTCW. The orthoMCL package has many steps to perform including one



Summary: (EaR >= 1 AND NnR >= 1 AND OlR >= 1 AND ShR >= 1), Method 'OrthoMCL_4'

Cluster View Filter 1: 5099 Clusters

Row #	Cluster ID	H-ID	H-Eval	H-Per	H-Desc	EaR	NnR	OlR	ShR
802	OM4_001009	Q2QZV1_ORYSJ	9E-90	28	Retrotransposon protein...	2	2	11	6
803	OM4_001010	PXM16_ARATH	0.0	100	Zinc-metallopeptidase, ...	5	4	8	4
804	OM4_001011	BH030_ARATH	2E-72	76	Transcription factor bHL...	2	9	3	7
805	OM4_001012	RHN1_NICPL	2E-96	76	Ras-related protein RH...	5	5	7	4
806	OM4_001013	E4MXZ6_THEHA	2E-179	80	mRNA, clone: RTFL01-3...	4	3	8	6
807	OM4_001014	Q10MI0_ORYSJ	0.0	23	Cyclin, putative, express...	4	4	6	7
808	OM4_001018	PDI14_ORYSJ	0.0	47	Protein disulfide isomera...	4	3	10	4
809	OM4_001019	B6SMS2_MAIZE	1E-61	23	Protein binding protein	5	5	4	7
810	OM4_001020	B9SJT7_RICCO	0.0	33	Importin beta-1, putative	7	3	6	5
811	OM4_001021	ARFR_ORYSJ	0.0	57	Auxin response factor 18	5	6	4	6
812	OM4_001022	G8A036_MEDTR	0.0	33	Protein tolB	2	10	4	5
813	OM4_001023	Vvi025522	6E-57	23	TF NF-YA	9	4	4	4
814	OM4_001024	PRS6A_SOLLC	0.0	47	26S protease regulatory ...	5	3	11	2
815	OM4_001025	Q5ZCR4_ORYSJ	0.0	66	Putative SAC domain pro...	4	3	2	12
816	OM4_001026	OPD23_ARATH	0.0	90	Dihydrolipoyllysine-resid...	4	3	11	3
817	OM4_001027	G7IDA8_MEDTR	1E-78	19	IST1-like protein (Fragm...	12	2	5	2
818	OM4_001028	SPIN1_ORYSJ	3E-124	95	KH domain-containing p...	4	6	6	5
819	OM4_001029	RF2B_ORYSJ	2E-101	95	Transcription factor RF2b	7	2	6	6
820	OM4_001030	Q5N7Q7_ORYSJ	0.0	42	(Rice Genome Annotatio...	12	2	4	3
821	OM4_001031	RAD23_ORYSJ	7E-165	42	Probable DNA repair pr...	5	4	8	4
822	OM4_001032	BSL2_ORYSJ	0.0	76	Serine/threonine-protein...	7	6	5	3
823	OM4_001033	DXS_CAPAN	0.0	95	Probable 1-deoxy-D-xy...	7	3	5	6
824	OM4_001034	Vvi030311	3E-31	23	TF GATA	12	3	3	3
825	OM4_001035	UMPK_ARATH	7E-82	100	UMP/CMP kinase	5	4	5	7
826	OM4_001036	B7EKF8_ORYSJ	2E-125	28	cDNA clone:J013135P1...	5	4	7	5

Figure 5. The viewMultiTCW interface. This shows the table of orthoMCL inflation 4.0 clusters where there is at least one sequence from each species. The H-ID and H-Desc columns are the annotation hit that the majority of the sequences have, e.g. for cluster OM4_0001010, 100% have the hit PXM16_ARATH where the best of them has an e-value of 0.0. The columns can be changed by selecting the "Select Columns" at the bottom, which will immediately change the table (i.e. a new search of the database is not required).
doi:10.1371/journal.pone.0069401.g005

that creates and uses a MySQL database; all steps are performed by runMultiTCW using the user-supplied value for the inflation parameter. The only drawback of including external programs in the TCW packages is that users may neglect to include their respective references in publications that use software within the TCW, hence, the TCW user agreement requires referencing TCW plus all other programs used for the analysis that are contained in the package.

R packages. These cannot be easily included in the package, but can all be easily installed with Bioconductor [34]. The TCW package provides instructions to step the user through the installation, which is only necessary if the user desires the DE calculations.

Command line. RunSingleTCW writes a configuration file that is used by the loader, assembler, and annotator executables. These programs can be run from the command line once the configuration file is created.

Demo and Help. Three single demos are available, as follows: one with Sanger ESTs plus Illlumina transcripts to assemble, one with transcripts and three libraries with three replica biological samples, and one with transcripts and two libraries with no replicas. These three databases can then be used to create a multiTCW database. All necessary files are contained in the package (e.g. the subset of the UniProt files for annotation) with the exception of those needed for DE and GO analysis, where the R packages would need to be installed and the GO tree downloaded, respectively. All interfaces have a "help" button to provide information about the interface. There is also a tour of the interfaces at http://www.agcol.arizona.edu/software/tcw/tour.

Interaction with other programs. ViewSingleTCW outputs multiple types of files from the table of query results: (1) all columns from the table as a CSV file, (2) the sequences from the table as a FASTA file (e.g. used as input to Blast2GO), and (3) the Pearson Correlation Coefficient will be computed for co-expression analysis using the RPKM columns (optionally log2 of the RPKM) in the table. ViewMultiTCW will export the contents of any table.

As a general way to add information into the singleTCW database, the user can simply import a file of 'seqID <remark>' rows, which can be viewed and searched on in the 'Remark' column.

Results

Results for the assembler are provided in Soderlund et al. [20]. Results are provided here using a dataset from the rhizome project (He et al. submitted) on timing and memory, as detailed in Table 1. The dataset used is from red rice (*Oryza longistaminata*) with 143,625 contigs and read count files for two paired end libraries plus 4 single-end libraries with 5 replicas each (22 total). The transcripts are annotated against the following 13 databases: plantTFDB [35], 5 taxonomic SwissProt and TrEMBL databases, and a SwissProt and TrEMBL subset of the entire UniProt (i.e. minus the 5 taxonomic and bacteria databases). The overall time required to perform the blasts of these 13 databases was a little over 6 days. Loading them into the TCW database along with the GC/ORF analysis and adding GO annotation took ~14.5 hours.

Table 2 shows the time for building a multi-species database from four rhizome species with the same libraries, and all have approximately the same number of sequences except for one that has a genome sequence, in which case, the gene models were used. The overall time was 7 hours, where the majority of that time was for the self-blast (on 24 CPUs).

Table 2. Timing results for building a multiTCW database from 4 singleTCW databases.

MultiTCW	Time	Remarks
Load 4 databases	33 m	431,888 transcripts
ESTscan[1]	8 m	170,311 protein[1]
Run selfblast	5 h:40 m	Using 24 CPUs
Run orthoMCL	18 m	18,184 clusters
Run transitive closure	1 m	7,767 clusters
Total	7 h:03 m	MySQL database 1.6G

[1] ESTscan does not produce protein sequences for all transcripts.

doi:10.1371/journal.pone.0069401.t002

Discussion

TCW uses MySQL as its database for a number of reasons. First, it is a common database and a biology laboratory is likely to already have it, which keeps installation simple. Second, orthoMCL uses MySQL, and the GO database is also available as MySQL tables, so there is no need for multiple database platforms. Third, TCW is meant for use by a single project and MySQL can easily handle the amount of data for a single laboratory analysis, i.e. it does not need to scale to sizes envisioned for data centers (e.g. iPlant). For example, a TCW project the size of red rice completely annotated (see Table 1) takes about 2.4G, which is just 0.24% of a terabyte hard drive.

The TCW uses common file formats to keep it simple, plus there is no need for more complicated file formats. Though an objective of the TCW is to reduce the use of spreadsheets, they will continue to be of value. For example, the biologist can export the contents of a table in delimited format, where Excel or a similar program can be used to produce graphs for publication. A second output is the FASTA file, where a filtered set of sequences can be used for analysis in a different program. The third output is the correlation coefficients of the DE values for use in pathway programs. These three file formats support analysis commonly used by individual laboratories.

De novo transcriptome assembly is much more compute-intensive than genome-assisted assembly. In both cases, blasting the transcripts against known protein or nucleotide databases is generally also compute intensive. To de novo assemble and map a project the size of red rice (4133M paired ends for assembly) took the following times (personal communications, Min-Jeong Kim, Gang laboratory): ~24 h for sequence cleaning with CLC Workbench (www.clcbio.com), 12–24 h for the first round assembly with CLC, ~1 h for polyA trimming with EMBOSS TrimEST [36], 12–48 h for GapClosing, ~24 h for the second round of assembly with iAssembler [37], and 1–2 h for mapping a single library. Therefore, it took ~6 days on modest size computers (12-CPU, 32G Ram, 1.5TB HDD Windows machine for CLC and an 8-CPU, 24G Ram, 2TB HDD Linux machine for all other steps). As shown in Table 1, the annotation blasts took about 7 days on a 24CPU processor (only the blast is parallelized). If a lab has a multi-processor computer, a project the size of red rice can be processed in about 13 days (albeit, UniProt has grown since the Apr-2012 release), and then the team members can use the computer for querying the results. Conversely, the availability of cloud computing will increase, and assembly and blasts are great applications for such computing, where the results can be used as input into TCW.

An important aspect of using a program such as TCW, which could reside on the project machine, is to encourage biologists to have a central location for all files and results. This is in contrast to various members of the team having files on their personal computers, which can cause loss of data. The project machine can contain all of the downloadable programs that the members need, e.g. TCW, MeV, CytoScape, Blast2GO. Since many bioinformatics tools run most conveniently on Linux, it is helpful if the team members know basic Unix commands, but these really are simple and can be learned in an hour or two. Users can log in directly from MACs, or from PCs using approaches such as VNC (Virtual Network Computing, www.realvnc.com). Using this approach, the team members have a shared environment with highly interactive graphics, providing a methodical approach for analysis.

Table 1. Timing results for building a singleTCW database.

SingleTCW	Time	Remarks
Load transcripts	12 m	143,625 transcripts plus 10 read count files
Instantiate transcripts	5 m	No assembly
Blast	6 d:5 h:00 m	13 databases[1] using 24 CPUs
Add hits and descriptions	13 h:50 m	Add 2M hits, 0.8M unique
Add GO terms	43 m	Add 4.7 k transcript GOs, 17 k unique GOs
DE with edgeR+FDR	9 m	One pairwise computation
Total	6 d:20 h:00 m	MySQL database 2.4G

[1] The plantTFDB database, the SwissProt and TrEMBL taxonomic databases for plants, invertebrates, fungi, viruses, vertebrate and a subset of the complete UniProt (i.e. minus the sequences from the bacteria and these 5 taxonomic databases). This is based on the April 2012 UniProt, where this set of databases (plus bacteria) take about 45G disk space.

doi:10.1371/journal.pone.0069401.t001

As a final word, in the past, many software programs required a high level of computer expertise to run along with the need to read pages of documentation to figure out what to do. The next-gen software programs provide, in general, easy one-step download and execution. For example, MAUVE [38], SyMAP [39], MeV, CytoScape, and TCW are easy to install Java programs, and Blast2GO and BioLayout [40] are webstart Java programs. All of these programs have interactive graphics for analysis and for viewing the results. It would provide a powerful dynamic approach for large-scale data and analysis of the 21st century to have this highly interactive environment working for laboratory projects, along with cloud computing for the large-scale processing and on-line databases to extract information.

Acknowledgments

We would like to acknowledge Min-Jeong Kim and Ruifeng He for creating the red rice dataset, Tonja Wilkins Fisher and Meenal Vyas for their feedback on the TCW, and Lomax Boyd for providing our system support.

Author Contributions

Conceived and designed the experiments: CS DRG. Analyzed the data: CS WN. Contributed reagents/materials/analysis tools: DRG. Wrote the paper: CS. Designed the software: CS. Software development: CS WN MW.

References

1. The GO Consortium (2012) The Gene Ontology: enhancements for 2011. Nucleic Acids Res 40: D559–564.
2. Garber M, Grabherr MG, Guttman M, Trapnell C (2011) Computational methods for transcriptome annotation and quantification using RNA-seq. Nat Methods 8: 469–477.
3. Conesa A, Gotz S (2008) Blast2GO: A comprehensive suite for functional analysis in plant genomics. Int J Plant Genomics 2008: 619832.
4. Hunter S, Jones P, Mitchell A, Apweiler R, Attwood TK, et al. (2012) InterPro in 2011: new developments in the family and domain prediction database. Nucleic Acids Res 40: D306–312.
5. Lohse M, Bolger AM, Nagel A, Fernie AR, Lunn JE, et al. (2012) RobiNA: a user-friendly, integrated software solution for RNA-Seq-based transcriptomics. Nucleic Acids Res 40: W622–627.
6. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26: 139–140.
7. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11: R106.
8. Risso D, Schwartz K, Sherlock G, Dudoit S (2011) GC-content normalization for RNA-Seq data. BMC Bioinformatics 12: 480.
9. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10: R25.
10. Wang L, Feng Z, Wang X, Zhang X (2010) DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. Bioinformatics 26: 136–138.
11. Young MD, Wakefield MJ, Smyth GK, Oshlack A (2010) Gene ontology analysis for RNA-seq: accounting for selection bias. Genome Biol 11: R14.
12. Li L, Stoeckert CJ Jr, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13: 2178–2189.
13. Ostlund G, Schmitt T, Forslund K, Kostler T, Messina DN, et al. (2010) InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. Nucleic Acids Res 38: D196–203.
14. Linard B, Thompson JD, Poch O, Lecompte O (2011) OrthoInspector: comprehensive orthology analysis and visual exploration. BMC Bioinformatics 12: 11.
15. Howe EA, Sinha R, Schlauch D, Quackenbush J (2011) RNA-Seq analysis in MeV. Bioinformatics 27: 3209–3210.
16. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, et al. (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15: 1451–1455.
17. Goff SA, Vaughn M, McKay S, Lyons E, Stapleton AE, et al. (2011) The iPlant Collaborative: Cyberinfrastructure for Plant Biology. Front Plant Sci 2: 34.
18. Goncalves A, Tikhonov A, Brazma A, Kapushesky M (2011) A pipeline for RNA-seq data processing and quality assessment. Bioinformatics 27: 867–869.
19. Langmead B, Hansen KD, Leek JT (2010) Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol 11: R83.
20. Soderlund C, Johnson E, Bomhoff M, Descour A (2009) PAVE: program for assembling and viewing ESTs. BMC Genomics 10: 400.
21. He R, Kim MJ, Nelson W, Balbuena TS, Kim R, et al. (2012) Next-generation sequencing-based transcriptomic and proteomic analysis of the common reed, Phragmites australis (Poaceae), reveals genes involved in invasiveness and rhizome specificity. Am J Bot 99: 232–247.
22. Huang X, Madan A (1999) CAP3: A DNA sequence assembly program. Genome Res 9: 868–877.
23. Dimmer EC, Huntley RP, Alam-Faruque Y, Sawford T, O'Donovan C, et al. (2012) The UniProt-GO Annotation database in 2011. Nucleic Acids Res 40: D565–570.
24. Tanabe M, Kanehisa M (2012) Using the KEGG database resource. Curr Protoc Bioinformatics Chapter 1: Unit1 12.
25. Bairoch A (2000) The ENZYME database in 2000. Nucleic Acids Res 28: 304–305.
26. Punta M, Coggill PC, Eberhardt RY, Mistry J, Tate J, et al. (2012) The Pfam protein families database. Nucleic Acids Res 40: D290–301.
27. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat Methods 5: 621–628.
28. Stekel DJ, Git Y, Falciani F (2000) The comparison of gene expression from multiple cDNA libraries. Genome Res 10: 2055–2061.
29. Hardcastle TJ, Kelly KA (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics 11: 422.
30. Smedley D, Haider S, Ballester B, Holland R, London D, et al. (2009) BioMart–biological queries made easy. BMC Genomics 10: 22.
31. Rhee SY, Wood V, Dolinski K, Draghici S (2008) Use and misuse of the gene ontology annotations. Nat Rev Genet 9: 509–515.
32. Iseli C, Jongeneel CV, Bucher P (1999) ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. Proc Int Conf Intell Syst Mol Biol: 138–148.
33. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32: 1792–1797.
34. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, et al. (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5: R80.
35. Zhang H, Jin J, Tang L, Zhao Y, Gu X, et al. (2011) PlantTFDB 2.0: update and improvement of the comprehensive plant transcription factor database. Nucleic Acids Res 39: D1114–1117.
36. Sarachu M, Colet M (2005) wEMBOSS: a web interface for EMBOSS. Bioinformatics 21: 540–541.
37. Zheng Y, Zhao L, Gao J, Fei Z (2011) iAssembler: a package for de novo assembly of Roche-454/Sanger transcriptome sequences. BMC Bioinformatics 12: 453.
38. Darling AC, Mau B, Blattner FR, Perna NT (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome Res 14: 1394–1403.
39. Soderlund C, Bomhoff M, Nelson WM (2011) SyMAP v3.4: a turnkey synteny system with application to plant genomes. Nucleic Acids Res 39: e68.
40. Theocharidis A, van Dongen S, Enright AJ, Freeman TC (2009) Network visualization and analysis of gene expression data using BioLayout Express(3D). Nat Protoc 4: 1535–1550.