Transcriptome computational workbench (TCW): analysis of single and comparative transcriptomes

Carol A. Soderlund
BIO5 Institute, University of Arizona, Tucson AZ, USA
Correspondence to cas1@u.arizona.edu

Abstract

De novo transcriptome sequencing and analysis provide a way for researchers of non-model organisms to explore the differences between various conditions and species. These experiments are expensive and produce large-scale data. The results are typically not definitive but will lead to new hypotheses to study. Therefore, it is important that the results be reproducible, extensible, queryable, and easily available to all members of the team. Towards this end, the Transcriptome Computational Workbench (TCW) is a software package to perform the fundamental computations for transcriptome analysis (singleTCW) and comparative analysis (multiTCW). It is a Java-based desktop application that uses MySQL for the TCW database. The input to singleTCW is sequence and optional count files; the computations are sequence similarity annotation, gene ontology assignment, open reading frame (ORF) finding using hit information and 5th-order Markov models, and differential expression (DE). For DE analysis, TCW interfaces with an R script, where R scripts for edgeR and DEseq are provided, but the user can supply their own. TCW provides support for searching with the super-fast DIAMOND program against UniProt taxonomic databases, though the user can request BLAST and provide other databases to search against. The input to multiTCW is multiple singleTCW databases; the computations are homologous pair assignment, pairwise analysis (e.g. Ka/Ks) from codon-based alignments, clustering (bidirectional best hit, Closure, OrthoMCL, user-supplied), and cluster analysis and annotation. Both singleTCW and multiTCW provide a graphical interface for extensive query and display of the data. Example results are presented from three datasets: (i) a rhizome plant with de novo assembled contigs, (ii) a rhizome plant with gene models from a draft genome sequence, and (iii) a non-rhizome plant with gene models from a finished genome sequence. The two rhizome plants have replicate count data for rhizomes, root, stem and leaf samples. The software is freely available at https://github.com/csoderlund/TCW.

Introduction

As the amount of transcriptome data increases from next generation sequencing, efficient and methodical computation is becoming more important. Given assembled sequences and read counts, there are various computations to be performed, i.e. similarity search, gene ontology (GO) [1] assignment, open reading frame (ORF) finding, and differential expression (DE) analysis. For comparative transcriptomes, the datasets can be compared using bidirectional best hit (BBH) and clustering algorithms, where the aligned pairs and clusters can be analyzed using various statistics. These analyses are typically computed with various downloaded programs, web-based programs, spreadsheets and custom scripts. This 'ad hoc' style of analysis can lead to lack of reproducibility, human error, loss of data and results, and is inadequate for extensibility.

The topic of reproducibility has been addressed in numerous publications. Peng [2] discusses how the introduction of computers to the analysis of biological data has introduced published computational results that are not reproducible. Stodden et al. [3] reviewed 204 *Science* publications and were able to reproduce the findings of only 26% of them. Stodden et al. [4] reviewed 170 journals policies on code data and code sharing. Garijo [5] reproduced the results of a published paper and quantified the difficulty; they concluded with reproducibility guidelines for authors, one of which was the importance of using published open source software whenever possible. Published open source software provides the details of how different computations are performed, and if there is any ambiguity, the software is available to determine the details. To compare results across publications, it is important that the analysis be equivalent, which can only be confirmed with access to the software or with detailed description.

The topic of reproducibility has drawn attention, but there are also the problems of loss of results, extensibility and accessibility. Bioinformatics often utilize many flat files (plain text file such as a FASTA file) of data and results, and it is complicated to keep such files organized. Computer scientists address this problem by using database management systems (DBMS), which store the data in one place and have fast update and retrieval algorithms. When an application specific interface to the DBMS is provided, any member of the research team can access the information, not just the person who performed the analysis. A DBMS also preserves the data and results for continued exploration after publication, making it easier to use the data in conjunction with future experiments.

Web-based applications have their benefits, but they should not be a substitute for the safe archival of data for a laboratory. They have two notable risks: (i) the lab is dependent on the host website to remain active and (ii) the results of a project could end up scattered across the network, or downloaded in many different flat files. Desktop applications help avoid these concerns, but they require a reasonably sized computer (see S1 Suppl §3) along with a compute savvy person who is capable of installing the necessary software and running the programs. However, computational analysis is permanently part of the biology world, so it should be routine for any lab that analyzes large-scale data to have a bioinformatics scientist as a member of the team.

The results of genomic software will typically not give "the" answer, but will provide evidence. This is the case with transcriptome analysis where the results provide evidence of function and expression, but typically do not provide an absolute answer. That means that query and display software should be an integral part of the biologist workbench so that the scientist has at their fingertips the ability to look at results from their data for evidence to be used in further experimentation and confirmation. Moreover, the biologist can get a better understanding of the data by viewing the details; for example, a BLAST [6] E-value of 1E-100 could be a long match with mismatches and gaps or a short exact match, where observing the actual alignment aids in understanding the characteristics of the match.

TCW is a desktop application that aids in creating reproducible and extensible results. It provides the fundamental computations for single transcriptome analysis (singleTCW) and for comparative analysis (multiTCW). They both use Java interfaces and the MySQL DBMS, which provides a central location for all data and results, making them easily accessible to all members of the research team. For singleTCW, the input is the sequence and optional replicate count files; the computation provided is sequence similarity annotation, ORF finding, GO assignment and differential expression. The user provides the annotation databases for the similarity search, where UniProt [7,8] is given special support. With the freely available super-fast DIAMOND [9], the searching is no longer a bottleneck. For the multiTCW, the input is two or more singleTCW databases; the computation provided is homologous pair assignment, clustering of homologous pairs, pairwise analysis and cluster analysis. The pairwise analysis is computed from the pairwise codon-based alignment, which produces statistics such as the number of synonymous and nonsynonymous codons, and Ka/Ks results [10]. The cluster analysis is from the multiple sequence alignments (MSA), which produces an alignment score and best annotation analysis. Both singleTCW and multiTCW provide interfaces for query and display, which allows the user to drill down to the details of input data and results.

There are numerous software packages for the upstream transcriptional analysis as reviewed in Poplawski et al. [11]. However, there is no other freely available open source software that provides the features stated above. MeV [12] is a desktop Java-based graphical interface for differential expression, but does not provide functional annotation. Blast2GO [13] is a desktop Java graphical interface for functional annotation, but the free version does not provide differential expression, nor is it open source. SATrans [14] provides analysis of the DESeq results, ORF finding and functional analysis, but does not provide graphical query and display. The S2 Suppl provides a more in-depth comparison of the available software for functional annotation and DE analysis. For comparative transcriptomes, there is no software that allows the in-depth study of the similarity between a few related transcriptomes. However, there are programs for clustering (e.g. OrthoMCL [15]) and multiple alignment (e.g. MAFFT [16]), which are used by multiTCW.

To demonstrate the TCW analysis, the software is applied to the transcriptomes of two rhizome plant species, the monocot red rice (*Oryza longistaminata*) and the eudicot sacred lotus (*Nelumbo nucifera*). The red rice contigs were de novo assembled [17] and the sacred lotus gene models were computed from the draft genome sequence [18]. In both cases, single-end Illumina sequences from rhizome, root, stem and leaf were aligned to the transcripts for quantification [17,19]. The NCBI transcript file from the non-rhizome cultivated rice *Oryza sativa* was also used in the comparison, where the gene models were computed from the complete genome sequence [20]. The term 'transcript' will be used for both the de novo contigs and the gene models.

Materials and Methods

TCW package

TCW has a long history, where it was originally developed for assembling Sanger ESTs [21], extended to assemble 454 data, and then further extended to take as input assembled sequences and count data [22]. Every aspect of TCW has been updated and many features added since the TCW v1 publication [22]. The following description is of the current state of TCW v3, where the workflow is shown in Fig 1. All examples are from the rhizome study, and a detailed description on how to reproduce them is given in the S1 Suppl. Details of the algorithms and supporting results are provided for singleTCW in the S2 Suppl and for multiTCW in the S3 Suppl.

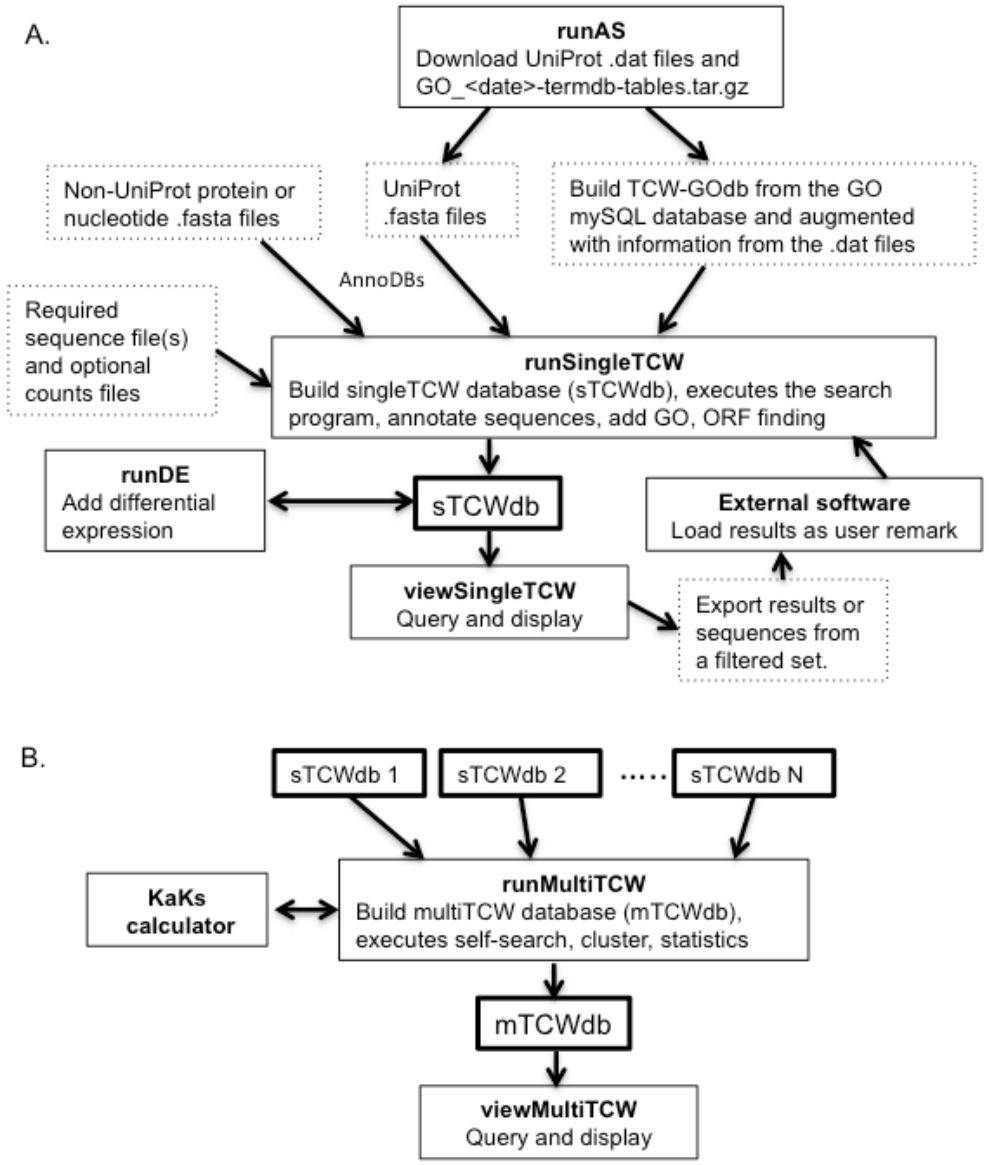

Fig 1. TCW programs and dataflow. (A) The steps to creating a singleTCW database. The only required input is the FASTA formatted sequence file(s). (B) The steps to creating a multiTCW database. The only required input is two or more sTCWdbs.

Input

TCW takes as input one or more FASTA files of nucleotide or amino sequences, where each FASTA file may have an associated file of quality values or tab-delimited file of counts. TCW can be used with transcriptomes with replicate count data, and can also be used with proteomes and replicate spectra (count) data. The ability to use TCW with both transcriptome and proteome data is useful when there are dual RNA-seq and peptide experiments, where the results can be equivalently analyzed with TCW; e.g. He et al. [23]. TCW has the capability of assembling Sanger ESTs, 454 and/or pre-assembled next generation transcripts. It can take as input location information, which is useful with gene models. Due to its multiple types of input, TCW generically refers to the input as "sequences with optional counts". The methods and results in this manuscript will focus on the input of transcripts with replicate counts. Table 1 shows the salient terminology for TCW.

Table 1. Terminology used by TCW.

TCW-GOdb	MySQL database created by runAS, which contains all GO terms and associated information from UniProt .dat files.
sTCWdb	MySQL database created by runSingleTCW, which contains the input sequences, counts and the analysis results.
mTCWdb	MySQL database created by runMultiTCW, which contains the data from multiple sTCWdbs and the comparative analysis results.
Sequence	The input sequences to TCW, e.g. transcripts, proteins, gene models.
annoDB	Annotation database, which is a file of protein or nucleotide sequences to search against.
Hit	A match between two sequences found by a search program (e.g. BLAST or DIAMOND).
Annotated sequence	A TCW sequence that has one or more hits to any annoDB.
Best Eval	The hit for a sequence that has the best E-value and bit score.
Best Anno	The best hit for a sequence that does not contain phrases such as "uncharacterized protein" (see S2 Suppl §2.1 for details).
Rank=1	The best E-value hit for a sequence to a given annoDB.

Installation and demo sets

TCW is available at https://github.com/CSoderlund/tcw. It has been tested on Linux and Mac. It requires Java, MySQL, BLAST and optionally R for DE analysis. For multiTCW, it optionally requires the KaKs-Calculator. The downloadable TCW package contains all other external software that it uses and the following demo sets: (i) input sequences to be assembled, (ii) protein sequences with replicate counts, (iii) nucleotide sequences with replicate counts, and (iv) three datasets that have good homology for input to multiTCW. The TCW software can be tried by downloading the package, untar'ing it, entering the MySQL information in the HOSTs.cfg file, and running it on the demo datasets. Step-by-step instructions are provided at www.agcol.arizona.edu/software/tcw along with additional information.

Build singleTCW

The runSingleTCW program provides a graphical interface to build the MySQL database (sTCWdb) with the sequence and count files, annotate the sequences and perform ORF finding. The first step is to load the sequences and counts into the database. The replicate counts are summed and the Reads Per Kilobase Million (RPKM) value calculated.

Sequence similarity annotations

Multiple annotation databases (annoDBs) can be used for sequence similarity, where an annoDB is a protein or nucleotide FASTA formatted file. TCW offers specific support for the UniProt [7,8] taxonomic databases with a graphical interface called runAS to download the desired UniProt data (.dat) files and create the corresponding FASTA file of sequences. TCW will also create a subset SwissProt, which is the entire SwissProt minus all sequences from the downloaded taxonomic databases. Fig 2 shows the annoDBs used for annotating the OlR dataset. The advantage of using the taxonomic databases is that the most relevant SwissProt and TrEMBL databases can be used and those hits can be queried by taxonomy using TCW.

ANNODB	ONLY	EVAL	ANNO	UNIQUE	TOTAL	AVG %SIM	HIT-SEQ (#SEQ)	BEST HIT AVG %SIM	COVER >=50	COVER >=90
SP-plants	199	5,252	10,594	23,274	133.0k	59.9	30,773 (21.4%)	64.7	27.3%	4.3%
SP-fungi	5	198	423	14,979	75,851	51.2	14,904 (10.4%)	48.9	14.9%	0.3%
SP-viruses	0	8	11	538	5,511	35.8	1,962 (1.4%)	37.0	2.8%	0%
SP-bacteria	0	18	24	32,799	74,518	45.8	7,471 (5.2%)	43.0	10.3%	0%
SP-invertebrates	3	459	568	9,226	58,857	52.8	14,315 (10.0%)	47.6	15.9%	0.2%
SP-fullSubset	6	167	277	28,001	131.4k	49.6	15,903 (11.1%)	47.0	14.6%	0.2%
TR-plants	27,088	54,982	47,115	541.1k	881.6k	85.4	68,586 (47.8%)	86.5	31.8%	9.3%
TR-fungi	2,250	7,757	7,278	276.5k	436.9k	53.6	28,126 (19.6%)	57.7	14.0%	1.2%
TR-viruses	1	16	9	8,159	41,709	39.7	5,802 (4.0%)	39.6	4.8%	<0.1%
TR-bacteria	126	457	685	177.1k	274.0k	46.5	21,069 (14.7%)	49.2	21.5%	1.0%
TR-invertebrates	229	1,647	1,775	216.1k	366.7k	50.4	28,838 (20.1%)	52.4	17.0%	0.5%
TF-PlantTFDB	187	2,531	4,733	30,127	80,598	67.2	20,112 (14.0%)	58.4	8.3%	1.5%

Fig 2. An overview of the annoDBs used to annotate the 143,625 OlR contigs. The following describes the columns. ANNODB: the "SP" and "TR" stand for SwissProt and TrEMBL, respectively. "TF" refers to the transcription factor database PlantTFDB [24]. The second part of the annoDB name is the taxonomy or source (named by the user). ONLY: the number of sequences that were only hit by the annoDB. EVAL and ANNO: the number of sequences from the annoDB assigned the Best Eval and Best Anno, respectively. UNIQUE: the number of unique identifiers from all sequence-hit pairs. TOTAL: the number of sequence-hit pairs from the annoDB. AVG %SIM: the average percent similarity for the total sequence-hit pairs. HIT-SEQ: the number and percent of sequences with at least one hit from the annoDB. BEST HIT AVG %SIM: the average percent similarity of the best E-value hit (Rank=1). COVER≥N: the percent of the HIT-SEQ that have similarity ≥ N% and hit coverage ≥ N% for the best E-value hit (Rank=1).

For similarity searching, TCW can use BLAST or DIAMOND. Given that DIAMOND is much faster than BLAST, it is the TCW default search program. When using the default parameters for either program, exact matches can be missed. TCW provides default parameters for DIAMOND, where they are set to maximize the number of perfect hits (see S2 Suppl §2.1.1). From all hits loaded for a sequence, a best E-value hit (Best Eval) and best annotation hit (Best Anno) will be computed; Fig 3 shows an example where the Best Eval hit is 'uncharacterized' and the Best Anno is "Microfibrillar-associated protein-related".

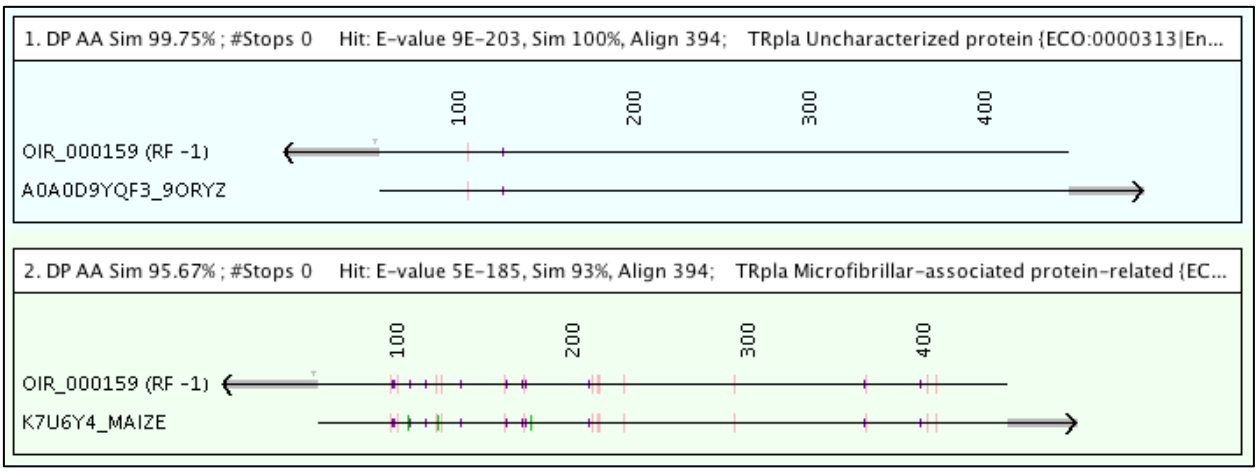

Fig 3. The best E-value hit (Best Eval) and best annotation hit (Best Anno) for OlR_000159. A dynamic programming algorithm is used for the alignment. The green marks are gaps, pink marks indicate BLOSUM ≤ 0, and the purple indicates BLOSUM > 0 where the substitution matrix is BLOSUM62 [25]. The gray areas at the ends are 'overhangs'.

GO assignment

The runAS interface is used to create the TCW GO database (TCW-GOdb), which contains the GO terms and their relations along with the UniProt GO assignments from the downloaded .dat files. The .dat files contain the sequence for each protein, the GO assignments with their evidence code, InterPro [26], KEGG [27], Enzyme Commission (EC) [28] and Pfam [29] assignments, where these data items are loaded into the TCW-GOdb. In runSingleTCW, after the sequences are annotated with hits from the UniProt annoDBs, the associated information from TCW-GOdb is assigned to each sequence. The InterPro, KEGG, EC and Pfam are direct assignments, but GO is more complicated since each direct GO assignment inherits all the ancestor GO terms. Therefore, each sequence-hit pair is assigned the direct GOs from the .dat file along with all inherited GOs. The GO Slims can be identified from a selected subset (http://www.geneontology.org/page/go-subset-guide).

As discussed by Rhee et al. [30], assigning levels to the GO terms is problematic because the GO structure is not uniform and each GO term can exist at multiple levels. Nevertheless, it is common to use levels since it provides the biologist an indication of where a given GO term is in the directed acyclic graph; therefore, TCW assigns the lowest level to each GO term. TCW also assigns the number of sequences with each GO term; this count is often used to show the GOs with the highest abundance of sequences, but as shown in the S2 Suppl §2.2.2, the count abundance of level 2 GO terms can lead to similar results across different species.

ORF finding and GC content

The TCW ORF finder algorithm primarily uses the ORF in the frame of the best hit with coordinates selected in relation to the hit alignment. If the sequence does not have a good hit, (i) the best ORF is found for each of the six reading frames and (ii) the best ORF from the six frames is selected. In both cases, the best ORF is selected as follows: use the longest ORF if the log length ratio > N, otherwise, use the ORF with the best Markov score. To compute the Markov score, the TransDecoder [31] algorithm was translated from Perl to Java. The sequence regions from the best hits are used to train the Markov model. The algorithm uses the hit coordinates for the ORF coordinates if the hit region has a valid start and stop codon, otherwise, it uses heuristics to determine how far to extend the region in search of a valid start and stop. ORF finding is complicated by transcripts with hits in multiple frames or hit regions with stop codons, so the algorithm uses heuristics for these cases. These heuristics along with details of the algorithm are discussed in S2 Suppl §2.3.

The computation of the GC content is performed on each sequence. For the overview, the average GC and CpG content are computed for the coding sequence (CDS) and untranslated regions (UTRs) derived from the computed ORF.

Differential expression

The TCW runDE program is used to compute the differential expression from the replicate counts and enter the results into the database. RunDE allows any R script to be used for the computation; it writes the necessary information to the R environment, runs the R script, and loads the results into the database. The R script for edgeR [32] and DEseq [33] are provided; to use either of these, the corresponding R code must be installed. The user can select to have the sequences pre-filtered based on the counts per million (CPM) or the raw counts. The CPM filter removes sequences that do not have CPM > N for ≥ M samples, where the CPM is the (count/sample size) * 1E06. The count filter removes sequences that do not have any sample with count > N. The results are entered into the sequence table of the sTCWdb with a user-supplied DE column name.

The TCW runDE program provides the ability to run GOSeq [34] on a DE column. GOSeq detects GOs that are over-represented based on a binary vector representing the sequences with DE p-values < N (default 0.05) along with the sequence lengths.

Other features

There is an option to compare all sequences, where the initial comparison uses BLASTn and/or tBLASTx, and the highest scoring N pairs are aligned using dynamic programming. This feature is especially useful for evaluating de novo assembled contigs for highly similar sequences.

Transcriptome and proteome publications generally have additional computations that are problem-specific; for example, transcripts are often analyzed for simple repeats. To use data and results from the TCW database, a file of results can be exported from viewSingleTCW for input to other programs, and then the external results can be imported into sTCWdb as user remarks using runSingleTCW. The user remarks can be searched and viewed in viewSingleTCW. Additionally, location data can be entered into the database for display.

Build multiTCW

The runMultiTCW program takes as input two or more sTCWdbs and builds the database (mTCWdb) (Fig 1B). Though there is no upper limit on the number of sTCWdbs to be compared, it is not meant for a large number (i.e. it has been used for up to four sTCWdbs). The input sTCWdbs can be built from nucleotide or protein sequences. The following will discuss an mTCWdb build from nucleotide sTCWdbs. When the mTCWdb is built, the nucleotide sequences, ORF coordinates, translated ORFs, normalized counts, DE, top hits and associated GOs are transferred from the sTCWdbs to the mTCWdb. The GC and CpG content is computed for the nucleotide sequence, and CpG Obs/Exp [35] is computed for the CDS and UTRs, where the CDS and UTRs are derived from the TCW computed ORF.

Pairs and clustering

A self-search of all amino acid (translated ORFs) sequences is performed along with an optional self-search of the nucleotide sequences. Either DIAMOND or BLAST can be used for the amino acid comparison; BLAST is required for the nucleotide comparison. The TCW BLAST defaults use soft-masking and the Smith-Waterman option, where Moreno-Hagelsieb and Latimer 2007 [36] showed that these options provide superior self-BLAST results. S3 Suppl §3.1 shows that DIAMOND produces comparable results to BLAST.

The self-search tab files are parsed, and the pairs are loaded into the database. The pairs are used as input to the clustering algorithms, where there are four options: (i) BBH (bidirectional best hit), (ii) Closure, (iii) OrthoMCL [15], and (iv) user-supplied. The first and second options are implemented within TCW and both have the following two parameters: minimum percent similarity and minimum percent coverage over one or both sequences. The third option executes the OrthoMCL code from within runMultiTCW and loads the results; this option has the one OrthoMCL parameter of 'inflation'. The last option allows the user to provide a file of clusters.

BBH (also referred to as Reciprocal Best Hits - RBH) is a common approach to use. Since more than two sTCWdbs may be compared, TCW provides N-way BBH, which first computes the 2-way BBH and then combines N-way BBH pairs. Alternatively, the user can select 2 datasets to use as input to the BBH algorithm. The Closure algorithm seeds the clusters with BBH hits, and then adds all sequences that (i) have a hit and (ii) pass the similarity and coverage rule with every other sequence in the cluster. OrthoMCL is a more sophisticated algorithm that builds a similarity matrix to normalize by species and uses Markov clustering, resulting in clusters that can have sequence pairs that are not in the hit file.

Annotation of clusters

After creating clusters, runMultiTCW annotates the clusters with the majority hit; that is, it finds the best common description substring among the sequences of the cluster and then finds the most common hit identifier for that description. The percent of sequences in a cluster with the description substring is computed.

The sequences of the clusters are aligned using the MSA program MAFFT [16]. The Trident [37] score, which considers the residue frequency and similarity per column, is computed for each cluster. The mStatX program (https://github.com/gcollet/MstatX) is used for the Trident calculation.

Annotation of pairs

Annotation is accomplished through two steps. First, given that the input sTCWdbs have the same conditions, the Pearson Correlation Coefficient (PCC) is calculated on the RPKM values between each pair of sequences. Second, runMultiTCW provides statistics for the pairs found in clusters that have a hit. Each pair is aligned using a dynamic programming algorithm of the two translated ORFs and then maps the results to the corresponding codon-based ORFs, resulting in a codon-based alignment. The statistics detailed in Table 2 are computed from each pair alignment; the description column of the table states whether gaps are included, but no statistics include the overhangs (see Fig 3). The aligned pairs are written to file and a shell script is written for the user to run the KaKs_Calculator on the pairs. The shell script specifies the method name "YN" [38], which the user can change to another method provided by the KaKs_Calculator. The Ka/Ks results are read into the mTCWdb.

Table 2: Pair statistics.

Column	Description[a]
5diff	% aligned bases in the 5'UTR that are different, includes gaps
3diff	% aligned bases in the 3'UTR that are different, includes gaps
Cdiff	% aligned bases in the CDS that are different, includes gaps
SNPs	# base differences in the CDS, excludes gaps
Gap open	# gap opens in the CDS (open = start of string of gaps)
Gap	# gaps in the alignment in the CDS
Align	# aligned bases in the CDS, includes gaps
Cov1	% of the 1st CDS that is covered with aligned bases
Cov2	% of the 2nd CDS that is covered with aligned bases
Calign	# aligned codons (amino acid characters), excludes gaps
Cexact	% aligned codons that are exact matches
Csyn	% aligned codons that are synonymous
C4d[b]	% aligned codons that are 4-fold degenerate
C2d[b]	% aligned codons that are 2-fold degenerate
CnonSyn	% aligned codons that are nonsynonymous
Aexact	% aligned amino acids that are exact matches
Apos	% aligned amino acids that are substitutions with BLOSUM62 score > 0
Aneg	% aligned amino acids that are substitutions with BLOSUM62 score ≤ 0
CpG	Jaccard index (#CpG in both codons)/(#CpG in either codon)
GC	Jaccard index (#C + #G in both sequences)/(#C + #G in either sequence)
ts/tv	ts=transition, tv=transversion
Ka[c]	Nonsynonymous substitution rate
Ks[c]	Synonymous substitution rate
KaKs[c]	Selective strength (< 1 purifying, = 1 neutral, > 1 positive)
p-value[c]	Fisher exact test of KaKs value

[a] All but the first two statistics are scored from the CDS codon-based alignment. The descriptions specify whether gaps are included, however, no statistics include the overhang (e.g. the gray regions at the ends of Fig 3).
[b] 4-fold and 2-fold degenerate are computed according to Lehmann and Libchaber [39]; N-fold are synonymous codons with N possible bases in the ith position.
[c] Calculated by the KaKs_Calculator [10] using the method specified by the user (default "YN").

TCW query and display

ViewSingleTCW provides filters and displays of the sTCWdb content. Briefly, the user can filter on the data associated with the sequences (e.g. Best Eval, DE, etc.), which results in a table of sequences (see S1 Suppl Figs S5-S6. The hits can be filtered, which results in a table of hits (see S1 Suppl Fig S4). The GOs can be filtered, which results in a table of GOs (see S1 Suppl Fig S7). The relations between the GOs, hits and sequences are complicated; to aid in understanding the data, TCW provides various views of the data (e.g. see S2 Suppl Figs S8B and S9B). For both the hit and GO tables, the associated sequences can be view in the sequence table. From the sequence table, a sequence can be selected to show all information associated with it (see S1 Suppl Fig S2), including its alignment to hits (Fig. 3 and S2 Suppl Figs S3-S6, S11).

ViewMultiTCW provides filters and displays of the mTCWdb content. Briefly, the user can filter on sequences, pairs and clusters resulting in corresponding tables (see S1 Suppl Figs S9-S15). The sequences in a pair or cluster can be viewed as a sequence table. The sequence table allows selected sequences to be pairwise aligned by nucleotide (full sequence, CDS, 5'UTR, 3'UTR) or amino acid (see S3 Fig S4A). From the graphical alignment panel, the text alignment can be viewed with annotation (e.g. the CpG sites, see S3 Suppl Fig S6). A cluster can be aligned by amino acid using MUSCLE [40] or MAFFT [16], or by nucleotide (full sequence or CDS) using MAFFT (see S3 Suppl Fig S2).

Both viewSingleTCW and viewMultiTCW produce overviews of their results and processing information (e.g. the dates of the GO tables and UniProts used), which is the initial view when starting either program. For all tables of results described for both viewSingleTCW and viewMultiTCW, the user can select the columns to view, move columns, and sort columns. Most columns can be filtered. All tables provide statistics on the selected numeric columns. All tables can be copied and exported in various formats. TCW does not provide graphical plots, but data can be exported to a tab-delimited file for input to a program that produces plots such as Excel.

Building the rhizome study databases

To demonstrate the TCW analysis, the software is applied to the transcriptomes of two rhizome and one non-rhizome plant species. (i) For the rhizome red rice (*Oryza longistaminata*), 143,625 contigs were de novo assembled from Illumina paired-end rhizome apical tip and elongation zone samples [17]. Single-end Illumina reads with 5 replicates from rhizome, root, stem and leaf were aligned to the transcripts for quantification [17]. (ii) For the rhizome sacred lotus (*Nelumbo nucifera*), 26,685 gene models were computed from the draft genome sequence [18]. As with red rice, single-end Illumina reads with 5 replicates from rhizome, root, stem and leaf were aligned to the transcripts for quantification [19]. (iii) For the non-rhizome model organism cultivated rice (*Oryza sativa),* 28,392 gene models were computed from the complete genome sequence [20]. The rhizome *O. longistaminata* and *N. nucifera* datasets are published datasets [17,19] and the *N. nucifera* and *O. sativa* gene models were downloaded from NCBI; see S1 Suppl §1 for the details.

The TCW databases were built as follows: The runAS program was used to download and process the SwissProt and TrEMBL taxonomic databases for plants, fungi, viruses, bacteria and invertebrate along with the SwissProt full database on 05-Dec-18. The TCW-GOdb was built from the downloaded file go_201812-termdb-tables.tar.gz. The runSingleTCW program was run to build sTCW_rhi_OlR (*O.*

longistaminata Rhizome), sTCW_rhi_NnR (*N. nucifera* Rhizome) and sTCW_Os (*O. sativa*). For OlR and NnR, the runDE program was executed to add the DE and GOseq results, where the DE was computed with the edgeR script and the CpM filtering defaults. The runMultiTCW program was run to build mTCW_rhi from the three sTCWdbs. Clusters were created using Closure, OrthoMCL and BBH, where TCW defaults were used for all computations.

The databases were built on a Linux machine with 24-core, 128 Gigabytes of RAM and download speed of 580 Mbsp. The databases were transferred to a Mac for interactive analysis. The time and memory used for building the full sTCW_rhi_NnR and mTCW_rhi databases on Linux are presented in the S1 Suppl §3. Additionally, timing results are provided for building sTCW_rhi_NnR on a Mac with 4-core, 16 Gigabytes and download speed of 50 Mbsp.

Results

Transcriptome analysis

Table 3 provides summary statistics of the three datasets. Both *Nelumbo nucifera* and *Oryza sativa* have proteins in TrEMBL, so the high percentage of annotated transcripts for NnR and Os was expected. OlR had only 51.2% hit transcripts even though it is closely related to *O. sativa*, which is likely due to problematic contigs from the assembly. S2 Suppl §2.1.3 shows graphs of the highest hitting species to Os and OlR, where the third highest hit to OlR was to the plant fungus pathogen *Gaeumannomyces graminis*. In fact, 8.6% of the transcripts have at least one hit to the fungal taxonomic database with E-value < 1E-30; as observed by He et al. [17], fungal genes may play important roles in rhizome tissue. All three sTCWdbs had Best Eval hits to the transcription factor sequences in PlantTFDB, where OlR had 2531, NnR had 1510, and Os had 907.

Table 3. Transcript and hit statistics

	#Trans	Average length	Hit[a]	Un-char[b]	TrEMBL plants			
					Hit[c]	AvgSim[d]	Cover≥50[e]	Cover≥90[e]
OlR	143,625	702	51.2%	25.3%	47.8%	86.5	31.8%	9.3%
NnR	26,685	1,467	97.6%	9.4%	97.4%	92.5	80.3%	55.3%
Os	28,392	1,738	99.9%	4.5%	99.9%	97.6	93.7%	80.7%

[a] The percent transcripts with at least one hit to any annoDB.
[b] The percent transcripts with a Best Anno of 'Uncharacterized protein'.
[c] The percent transcripts with at least one hit to the TrEMBL plant annoDB.
[d] Average similarity of the best hit to the TrEMBL plant annoDB.
[e] Cover ≥ N is the percent transcripts that have a best hit with coverage ≥ N and similarity ≥ N.

The average length of the ORFs for OlR, NnR and Os was 336, 1104 and 1143 nucleotides, respectively. Table 4 provides a summary of their composition. Finding the longest ORF often computes the correct ORF, but not always; for example, the Os database had 1241 ORFs that were not the longest, but were exactly aligned to the hit with an ATG and stop codon at the ends. ORF finding is complicated when there are multiple hit frames or stop codons within a hit; the percentages of these two cases (multiple frames, stop codons) were OlR (6.0%, 6.2%), NnR (3.0%, 2.6%) and Os (12.7%, 9.6%). Interestingly, the Os transcripts, which are gene models from the complete genome, have the highest numbers of multiple frames and stop codons; these occurrences can be easily viewed in TCW, which is illustrated in S2 Suppl §2.1.2.

Table 4. Summary of ORFs

All ORFs[a]	OlR	NnR	Os	Good coverage hit[b]	OlR	NnR	Os
% ORF frame=hit frame	51.2	97.4	99.9	% of transcripts	9.1	58.8	79.6
% Hit & Longest & Markov	28.6	92.6	81.4	% Longest & Markov	77.3	98.2	83.6
% Longest ORF	60.8	95.8	88.2	% Longest ORF	83.1	98.9	89.4
% Markov best score	62.7	96.8	89.8	% Markov best score	89.4	99.1	91.5
% Markov good frame	54.8	95.7	92.0	% Markov good frame	91.5	98.1	92.3
% Has start & stop	29.8	81.1	70.4	% Has start & stop	75.4	91.5	83.6
% ORF=Hit & ends[c]	3.1	53.1	65.1	% ORF=Hit & ends[c]	58.3	84.6	80.9

[a] Percent transcripts, which all have ORFs (OlR: 143,625; NnR: 26,685; Os: 28,392).
[b] Percent ORFs with a good coverage hit (OlR: 6752, NnR: 15,311; Os: 22,592), with the exception that the first row is the percent of all ORF. A good coverage hit requires that 95% of the hit is covered and the hit has 60% similarity with no stops in the hit region.
[c]The ORF coordinates are the same as the hit coordinates and end exactly with an ATG and stop.

To explore the differential expression, the NnR database was queried for transcripts that were preferentially expressed in the rhizome compared to the other tissues. Using p-value < 0.0001, there were 584 up-regulated transcripts and 825 down-regulated transcripts when comparing rhizomes to root, stem and leaf. Of the 584 up-regulated transcripts, 153 had RPKM ≥ 100 for rhizome. Of the 825 down-regulated transcripts, 34 had RPKM ≥ 100 for root, stem and leaf. The top up-regulated transcripts are shown in the TCW sequence table in S1 Fig S6.

NnR was queried for the level 3 biological process GOs, which resulted it 141. It was then queried for the over-represented level 3 biological process GOs that were differentially-expressed for RhRo (rhizome-root), RhSt (rhizome-stem) or RhOL (rhizome-old leaf) using GOseq p-value < 0.001. From the total 141, 21 of them were DE for RhRo, RhSt or RhOL (Fig 4). S2 Suppl §3.2.1 shows the GO results for biological processes that were DE for RhRo, RhSt and RhOL using REVIGO [41], WEGO [42] and the TCW trim algorithm [43].

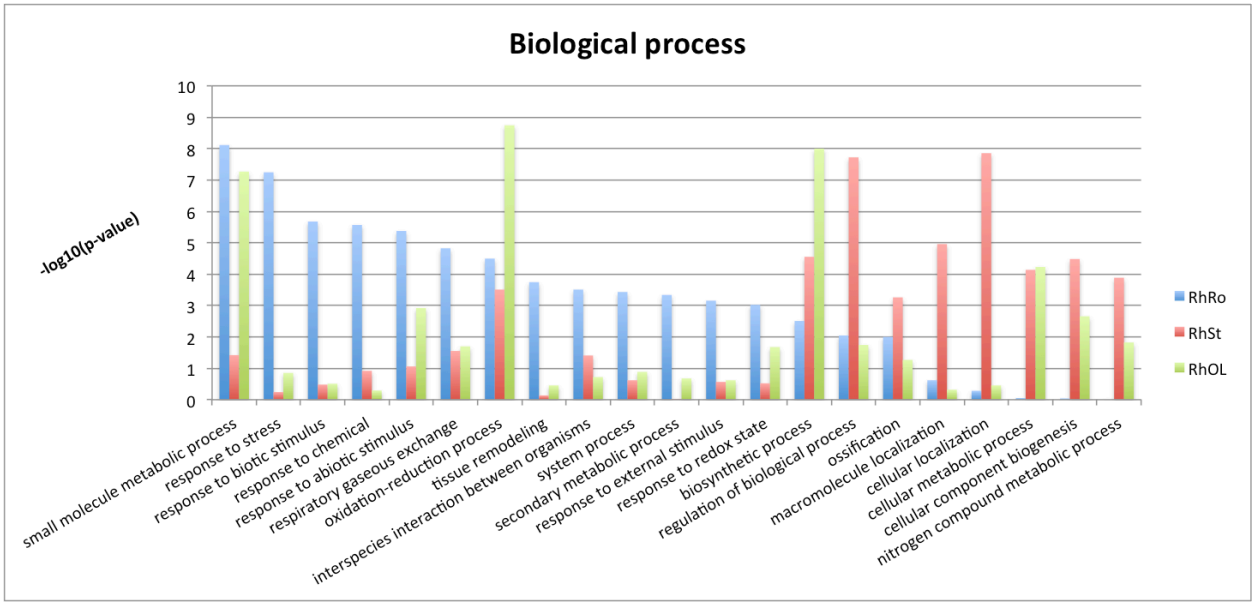

Fig 4. GO enrichment of biological process level 3 GO terms for NnR rhizomes. Rhizomes were compared to root (RhRo), stem (RhSt) and leaf (RhOL) using p-value < 0.001.

Comparative transcriptome analysis

The mTCW_rhi comparative database was created from the OlR, NnR, and Os sTCWdbs. Table 5 displays the number of clusters for BBH, Closure and OrthoMCL. BBH and Closure used the default cutoffs of similarity ≥ 60% and coverage ≥ 40% for both transcripts. The OrthoMCL inflation parameter was set to 4. The BBH OlR-Os statistics support that the *Oryza* species are closely related with the high average Trident score and the low average nonsynonymous and synonymous substitution rates.

Table 5: Summary of clusters

Clusters							Cluster pairs[c]	
Method[a]	**=2**	**3-5**	**6-15**	**>15**	**#Seqs**	**Average Trident[b]**	**Avg Ka**	**Avg Ks**
OlR-Os	13,965	-	-	-	14.0%	0.74	0.024	0.356
NnR-OlR	4,455	-	-	-	4.5%	0.56	0.171	7.71
NnR-Os	4,903	-	-	-	4.9%	0.61	0.176	8.33
3-way	-	2,896	-	-	4.4%	0.65	0.122	4.63
CL	17,412	4,317	994	31	33.5%	0.60	0.124	8.51
OM	7,523	7,712	3,887	958	49.3%	0.25	0.273	12.25

[a] The first four sets are BBH, CL is Closure and OM is OrthoMCL.
[b] Trident scores are between 0 and 1 where 1 is the most conserved.
[c] Average Ka (nonsynomymous rate) and Ks (synonymous rate).

The OM clusters were filtered on having at least one transcript from each dataset, which resulted in 7443 clusters with an average Trident score of 0.2597. The same filter was applied to CL clusters, which resulted in 3832 clusters with an average Trident score of 0.5647. One of these Closure clusters, CL_004428, is shown in Fig 5A; this view shows that the first two aligned amino acid sequences are identical (prefix NM_ is an Os transcript), but their nucleotide sequences have multiple synonymous codons and the UTRs have gaps.

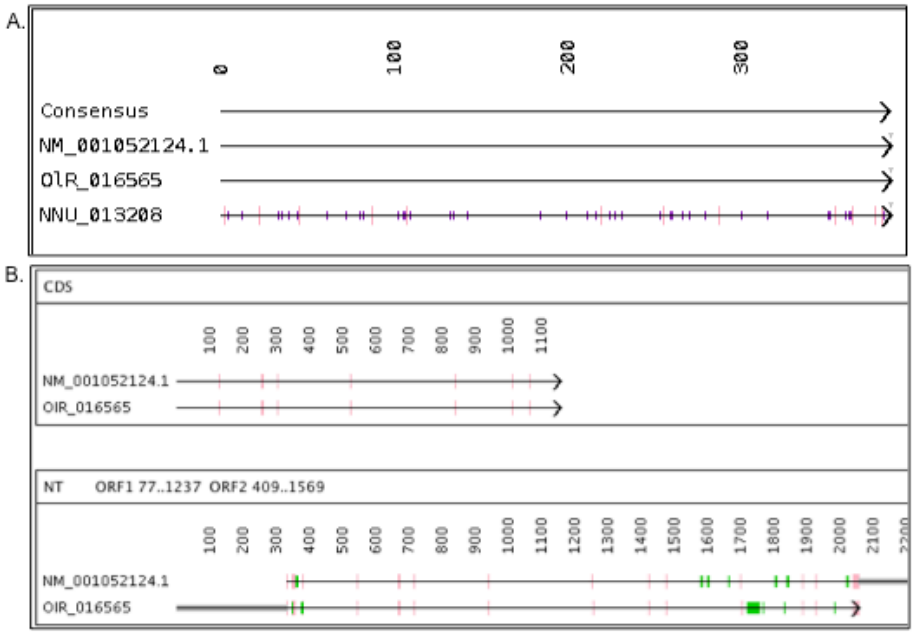

Fig 5. CL_004428 PHD glycerate dehydrogenase. (A) The MSA of a cluster where the first two amino acid sequences are identical. (B) For the identical amino acid sequences, the alignment of the CDS and full nucleotide sequence are shown. The pink marks are mismatches and the green marks are gaps.

Clusters that have at least one transcript from the two rhizome datasets OlR and NnR, and no transcripts from the non-rhizome Os dataset are potential rhizome-specific proteins. There were 510 candidate rhizome clusters in the OrthoMCL set, where all of them were annotated and 33 had the description of "uncharacterized". There were 867 candidate rhizome clusters in the Closure set, where all of them were annotated and 25 had the description "uncharacterized". Fig 6 shows the top 10 of the table of 867 Closure clusters sorted by Trident score. S3 Suppl §4.2.1 has a graph of the over-represented level 2 GOs for this set. Further analysis of these clusters would be of interest to a biologist studying rhizomes, where the descriptions and GO assignments could elucidate important functionality to rhizomes.

Filter: Set=CL, (OlR >= 1 AND NnR >= 1), (Os <= 0)

Cluster View 867 of 71339 (1%)

Row #	ClusterID	conLen	Trident	%Hit	E-value	Descript	Species	OlR	NnR	Os
1	CL_000156	1228	0.908	100	0.0	DNA-directed RNA poly...	Oryza glaberrima	1	1	0
2	CL_007497	304	0.809	100	8E-157	Ornithine carbamoyltran...	Dichanthelium olig...	1	1	0
3	CL_013854	68	0.809	100	4E-16	C2H2 family protein {kfl...	Klebsormidium fla...	1	1	0
4	CL_012762	125	0.794	100	1E-12	Uncharacterized protein ...	Ancylostoma ceyla...	2	1	0
5	CL_000023	2191	0.782	100	0.0	DExH-box ATP-depend...	Zea mays	1	1	0
6	CL_003771	518	0.780	100	4E-278	Putative ceramide glucos...	Oryza sativa subs...	1	1	0
7	CL_000847	1034	0.775	100	0.0	Protein translocase subu...	Oryza glaberrima	1	1	0
8	CL_003587	670	0.774	100	0.0	Ankyrin repeat protein-li...	Oryza sativa subs...	1	2	0
9	CL_006054	404	0.772	100	4E-179	Photosystem II stability/a...	Arabidopsis thalia...	1	1	0
10	CL_006558	371	0.766	100	4E-187	Cytosolic Fe-S cluster as...	Oryza sativa subs...	2	1	0

Fig 6. Ten annotated clusters specific to the rhizome datasets. The conLen column is the consensus length. The E-value, Descript and Species are from the best shared hit, and the %Hit is the percent of transcripts in the cluster with the description substring. The OlR, NnR and Os columns are the number of transcripts from each respective dataset.

Analysis of BBH clusters

Fig 7 shows the BBH average statistics from the related OlR-Os pairs and the more distantly related NnR-Os. The 93.9% exact codons for OlR-Os elucidates how closely related they are compared to the 34.8% for NnR-Os, though NnR-Os does have 74.1% exact amino acids.

BBH OlR-Os

Aligned: 13,965 CDS: 15.3Mb 5UTR: 1.2Mb 3UTR: 3.0Mb

Codons	5.0M	Amino Acids		Nucleotides		
Exact	93.9%	Exact	97.0%	CDS	Diff	5.2%
Synonymous	3.1%	Substitution >0	1.8%		Gaps	2.7%
Fourfold	1.9%	Substitution<=0	1.2%		SNPs	2.5%
Twofold	1.0%			5UTR	Diff	22.0%
Nonsynonymous	3.0%			3UTR	Diff	10.9%

BBH NnR-Os

Aligned: 4,922 CDS: 6.0Mb 5UTR: 361.2kb 3UTR: 912.7kb

Codons	1.8M	Amino Acids		Nucleotides		
Exact	34.8%	Exact	74.1%	CDS	Diff	36.0%
Synonymous	39.4%	Substitution >0	17.7%		Gaps	8.4%
Fourfold	21.4%	Substitution<=0	8.2%		SNPs	27.6%
Twofold	13.8%			5UTR	Diff	51.4%
Nonsynonymous	25.9%			3UTR	Diff	50.1%

Fig 7. Pair statistics from the BBH OlR-Os and NnR-Os clusters. The OlR-Os are more closely related than the NnR-Os. The explanation of each statistic is given in Table 2.

As stated by Zhang et al. [10], the selection is neutral if Ka=Ka, purifying if Ka<Ks, and positive (diversifying) if Ka>Ks. There is also the case where either Ka or Ks is zero. The Ka/Ks values were computed for all hit pairs that are in clusters, with the following number of pairs: 5522 zero; 0 Ka=Ks; 295,481 Ka<Ks; 2933 Ka>Ks. Fig 8A shows the Ka/Ks summary for the NnR-Os BBH pairs where only one pair is under purifying selection. For the 1230 pairs from the first quartile (KaKs < 0.03690), there were 50 with no description and 8 that were uncharacterized. For the 1230 pairs from the third quartile (KaKs ≥ 0.07874), there were 127 with no description and 58 that were uncharacterized. Fig 8B-8C shows the 10 pairs with the lowest and 10 pairs with the highest Ka/Ks scores, respectively.

A.

Average		Ka/Ks		Quartiles		P-value	
Ka	0.176	Zero	2	Q1(Lower)	0.03690	<1E-100	3,628
Ks	8.393	Ka=Ks	0	Q2(Median)	0.05368	<1E-10	1,265
P-value	0.000	Ka<Ks	4,919	Q3(Upper)	0.07874	<0.001	24
		Ka>Ks	1			Other	5

B.

Filter: KaKs<0.0369 AND In(Bns)

Pair View 1230 of 855697 (<1%)

Row #	SeqID1	SeqID2	Descript	%AAsim	Align	Ka	Ks	KaKs
1	NM_001060698.2	NNU_012333	PHD finger-like domain-containing protein ...	100.0	333	--	3.27	5.1E-12
2	NM_001056811.1	NNU_001819	Histone H3.3	100.0	411	--	3.28	1.3E-11
3	NM_001189859.1	NNU_025416	Ubiquitin-like protein 5	94.6	222	0.021	99.0	2.1E-04
4	NM_001050403.1	NNU_015607	60S ribosomal protein L37a-1	96.7	285	0.031	99.0	3.2E-04
5	NM_001064777.1	NNU_004087	Protein transport protein Sec61 subunit ga...	92.9	210	0.035	99.0	3.6E-04
6	NM_001053401.1	NNU_005163	40S ribosomal protein S8	92.2	669	0.037	99.0	3.8E-04
7	NM_001051874.1	NNU_010998	Photosystem II reaction center Psb28 protei...	91.3	276	0.048	99.0	4.8E-04
8	NM_001059845.1	NNU_017229	60S ribosomal protein L14-1 {ECO:00003...	90.2	405	0.048	99.0	4.9E-04
9	NM_001073616.1	NNU_013663	Mitochondrial import inner membrane trans...	89.4	282	0.048	99.0	4.9E-04
10	NM_001056406.1	NNU_016237	Anaphase-promoting complex subunit 11 {...	89.7	234	0.049	99.0	4.9E-04

C.

Filter: KaKs>=0.07874 AND In(Bns)

Pair View 1230 of 855697 (<1%)

Row #	SeqID1	SeqID2	Descript	%AAsim	Align	Ka	Ks	KaKs
1	NM_001066055.2	NNU_010947	-	67.0	1377	0.441	0.364	1.21
2	NM_001071264.1	NNU_015971	Glycine-rich cell wall structural protein 2	66.3	639	0.422	0.527	0.800
3	NM_001187395.1	NNU_006131	50S ribosomal protein L14, chloroplastic {EC...	76.1	213	0.125	0.435	0.288
4	NM_001063462.1	NNU_009911	36.4 kDa proline-rich protein {ECO:000031...	60.3	897	0.325	1.13	0.288
5	NM_001053936.1	NNU_021606	uncharacterized protein LOC104613166 {EC...	60.2	519	0.316	1.15	0.274
6	NM_001053710.1	NNU_024659	transmembrane protein 50 homolog {ECO:00...	63.9	414	0.351	1.28	0.273
7	NM_001050861.1	NNU_005174	Uncharacterized protein {ECO:0000313|Ense...	65.6	459	0.399	1.48	0.270
8	NM_001072099.1	NNU_001692	biogenesis of lysosome-related organelles co...	61.3	387	0.289	1.28	0.225
9	NM_001048563.1	NNU_005415	Metallothionein-like protein 2A	67.5	258	0.206	0.943	0.218
10	NM_001070418.1	NNU_012998	uncharacterized protein LOC104603547 {EC...	60.6	225	0.275	1.31	0.210

Fig 8. Ka/Ks for BBH NnR-Os. (A) The overview for the Ka/Ks results for the BBH NnR-Os pairs. (B) The ten pairs from BBH NnR-Os (Bns) with the lowest Ka/Ks scores. (C) The ten pairs from BBH NnR-Os (Bns) with the highest Ka/Ks scores. The 'Align' column is the number of aligned bases and the '%AAsim' is the identity score from the hit file.

Comparison with other programs

S2 Suppl §2.3.2 compares the TCW ORF finder results with the TransDecoder ORF finder [31]. From 2661 *O. sativa* transcripts where 68.8% had hits to SwissProt plants, the two programs agreed on 1772 ORFs, 730 had different start coordinates, 6 had different end coordinates, 5 had overlapping coordinates, and 102 sequence did not have an ORF predicted by TransDecoder. The difference in start coordinates was typically because TransDecoder selected coordinates for the longest ORF whereas TCW selected coordinates that best adhere to the hit region. The most significant differences are when the ORFs are translated into totally different amino acid sequences, which happens when there are non-overlapping coordinates or different frames; the significant differences between these two ORF finders were 9 ORFs with non-overlapping coordinates and 37 with different frame. These were compared with the full sTCW_Os, which was annotated with TrEMBL plants that contains the *O. sativa* proteins. The results showed that sometimes the TCW-computed ORF was correct and sometimes TransDecoder was. With current approaches, there are no perfect rules that call the ORF correctly 100% of the time.

In the S3 Suppl §4.1.1, the TCW BBH results were compared to Galaxy BBH [44], where there were at least four differences in implementation details. For example, TCW performs a self-search of all sequences whereas Galaxy searches the sequences from one dataset against the other. For a second example, when using the similarity cutoff of 70%, TCW will round up a 69.8% result whereas Galaxy does not. After removing differences that could have been easily fixed (e.g. use a cutoff of 71% in TCW), they shared 736 BBH pairs, TCW had 29 pairs that were not in Galaxy and Galaxy had 16 pairs that were not in TCW. This comparison demonstrates that subtle differences in algorithms impact the results. It also shows the problem with cutoffs, as making a cutoff too low introduces false positives and making it too high introduces false negatives; that is, there is no perfect cutoff. The BBH algorithm is a heuristic approach for determining approximately how many orthologs exist between two species; one cannot infer that the BBH pairs are strictly orthologous or that all orthologs are detected (see Koski and Golding [45] and Dalquen and Dessimoz [46]). By the very nature of the problem, there is no perfect set of rules.

Comparing results that are computed by different programs will typically provide dissimilar values because most computational genomic algorithms require some heuristics and cutoffs, which can vary from program to program. Though the difference in the results from the ORF and BBH algorithm comparisons are minor, the differences with in-house unpublished algorithms could be much more extensive. Authors of publications have a responsibility to ensure that their results can be compared with other similar experiments, either by using published software or providing adequate explanation of their in-house software.

In S2 Suppl §4, the singleTCW is compared with other programs, which have a range of functionality, none of which is the same as what TCW provides. Besides different functionality, they have different presentation of the results. Most of them provide graphs of the results, but as shown, by exporting the TCW results to a spreadsheet, programs such as Excel can easily make charts. Graphs alone do not provide a full understanding of the results, where interactive graphics are required to fully understand the data and results. The only two programs that provide extensive query and display are the proprietary Blast2GO and the open source TCW.

In regards to the comparative multiTCW, there is no published software to compare it to.

Discussion

TCW provides easily reproducible results. The supplements provide instructions on how to reproduce the results in the manuscript and the supplements. In addition to the value of reproducible results for this manuscript, a significant benefit is that researchers using TCW can produce the same type of results for their data. This saves time in figuring out how to generate the results and for writing detailed methods on how they were produced.

TCW provides flexibility in its processing. For singleTCW, the user can provide a tabular file of search results if they want to use a program other than BLAST or DIAMOND; the user provides the annotation databases, which can be protein or nucleotide; and an R script other than what TCW provides can be used for differential expression analysis. For multiTCW, clusters can be input from software other than what TCW provides, and the Ka/Ks values can be input from a program other than the KaKs_Calculator. Additionally, the design of the viewMultiTCW code makes it easy to add display columns and additional filters. This flexibility allows the user to try different approaches and view the results. It also provides developers of search algorithms, differential expression, cluster algorithms and Ka/Ks computation an interface to view their results.

TCW has two strengths, one is the functionality and advanced algorithms, and the other is the extensive query and display capabilities. TCW allows the user to view all the results, nothing is hidden, and the graphics allow the user to verify all results. For example, using viewSingleTCW the user can view over-expressed GOs for a given p-value, and can then drill down to view all sequences that have a hit with a given GO term, and then further view a given sequence with its hits and GOs. Another example is with viewMultiTCW, where the user can view the text alignment of a homologous pair along with the location of the different statistics listed in Table 2, such as the locations of the CpG sites. In summary, allowing the biologist to view the details of the computation aids in clearly understanding the results. It identifies where the ambiguities and difficulties are and demonstrates why there is not a perfect algorithm to solve many of these problems. This in turn could lead to refinement of the wet lab experiments, which produce data that result in more precise computational results.

Biologists explore transcriptomes through ingenious wet lab experiments; however, with large-scale data they must also be proficient at exploring the transcriptomes through computational approaches. Though there is exploratory software for other genomic problems, especially for human genomics, non-model organism large-scale genomic research would benefit from more exploratory software. The biologist should be able to have at their workbench a computer with multiple applications to aid in exploring this wealth of information. Though TCW only solves a subset of the problems involved in transcriptomes, and its interactive features could be further extended, it far surpasses anything else available. In summary, TCW provides a valuable and unique software package for transcriptome analysis.

Supporting information

S1 Suppl. Datasets, reproduce results, timings. Sections: (1) details of the datasets used in the manuscript and supplements, (2) instructions on how all results in the manuscript were obtained along with TCW snapshots, (3) timings of the builds, (4) major changes since TCW v1, and (5) future directions.

S2 Suppl. Build a singleTCW database. Sections: (1) using runAS for downloading UniProts and GO for TCW annotation, (2) the runSingleTCW annotation: default DIAMOND parameters, *O. sativa* multi-frame hits and hits with stop codons, graphs of species for Os and OlR, GO levels, multi-species level 2 graph, the ORF finding algorithm with comparison to TransDecoder, adding external data, (3) runDE for differential expression, GO results using REVIGO and WEGO, and (4) comparison with other transcript annotation software.

S3 Suppl. Build a multiTCW database. Sections: (1) the runMultiTCW interface, (2) building the database and adding GOs, (3) search parameters, (4) computing clusters, the BBH algorithm with comparison to Galaxy BBH, the Closure algorithm with a graph of the rhizome-specific over-represented GOs, OrthoMCL and Closure results with different parameters, (5) annotation of pairs and clusters, (6) GC, CpG, Ts/Tv computations, and (7) RPKM, DE and PCC in multiTCW.

Acknowledgments

The 24-CPU Linux machine used for building the databases is housed at the BIO5 Institute and Lomax Boyd of BIO5 provided the system maintenance. KS Riley edited the manuscript.

References

1. The Gene Ontology Consortium. Gene Ontology Consortium: going forward. Nucleic Acids Research. 2015;43: D1049–D1056. doi:10.1093/nar/gku1179

2. Peng RD. Reproducible Research in Computational Science. Science. 2011;334: 1226. doi:10.1126/science.1213847

3. Stodden V, Seiler J, Ma Z. An empirical analysis of journal policy effectiveness for computational reproducibility. Proc Natl Acad Sci USA. 2018;115: 2584. doi:10.1073/pnas.1708290115

4. Stodden V, Guo P, Ma Z. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. PLOS ONE. 2013;8: e67111. doi:10.1371/journal.pone.0067111

5. Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLOS ONE. 2013;8: e80278. doi:10.1371/journal.pone.0080278

6. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinformatics. 2009;10: 421. doi:10.1186/1471-2105-10-421

7. The UniProt Consortium. UniProt: a hub for protein information. Nucleic Acids Research. 2015;43: D204–D212. doi:10.1093/nar/gku989

8. Dimmer EC, Huntley RP, Alam-Faruque Y, Sawford T, O'Donovan C, Martin MJ, et al. The UniProt-GO Annotation database in 2011. Nucleic acids research. 2012;40: D565–D570. doi:10.1093/nar/gkr1048

9. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Meth. 2015;12: 59–60.

10. Zhang Z, Li J, Zhao X-Q, Wang J, Wong GK-S, Yu J. KaKs_Calculator: Calculating Ka and Ks Through Model Selection and Model Averaging. Genomics, Proteomics & Bioinformatics. 2006;4: 259–263. doi:10.1016/S1672-0229(07)60007-2

11. Poplawski A, Marini F, Hess M, Zeller T, Mazur J, Binder H. Systematically evaluating interfaces for RNA-seq analysis from a life scientist perspective. Brief Bioinform. 2016;17: 213–223. doi:10.1093/bib/bbv036

12. Howe EA, Sinha R, Schlauch D, Quackenbush J. RNA-Seq analysis in MeV. Bioinformatics (Oxford, England). 2011;27: 3209–3210. doi:10.1093/bioinformatics/btr490

13. Conesa A, Götz S. Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics. International Journal of Plant Genomics. 2008;2008: 619832. doi:10.1155/2008/619832

14. Kokas FZ, Bergougnoux V, Cudejkova MM. SATrans: New Free Available Software for Annotation of Transcriptome and Functional Analysis of Differentially Expressed Genes. J Comput Biol. 2018; doi:10.1089/cmb.2018.0149

15. Li L, Stoeckert CJ, Roos DS. OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. Genome Research. 2003;13: 2178–2189. doi:10.1101/gr.1224503

16. Katoh K, Standley DM. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. Molecular Biology and Evolution. 2013;30: 772–780. doi:10.1093/molbev/mst010

17. He R, Salvato F, Park J-J, Kim M-J, Nelson W, Balbuena TS, et al. A systems-wide comparison of red rice (Oryza longistaminata) tissues identifies rhizome specific genes and proteins that are targets for cultivated rice improvement. BMC Plant Biology. 2014;14: 46. doi:10.1186/1471-2229-14-46

18. Ming R, VanBuren R, Liu Y, Yang M, Han Y, Li L-T, et al. Genome of the long-living sacred lotus (Nelumbo nucifera Gaertn.). Genome Biology. 2013;14: R41. doi:10.1186/gb-2013-14-5-r41

19. Kim M-J, Nelson W, Soderlund CA, Gang DR. Next-Generation Sequencing-Based Transcriptional Profiling of Sacred Lotus "China Antique." Tropical Plant Biology. 2013;6: 161–179. doi:10.1007/s12042-013-9130-4

20. International Rice Genome Sequencing Project, Sasaki T. The map-based sequence of the rice genome. Nature. 2005;436: 793.

21. Soderlund C, Johnson E, Bomhoff M, Descour A. PAVE: Program for assembling and viewing ESTs. BMC Genomics. 2009;10: 400. doi:10.1186/1471-2164-10-400

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

23. He R, Kim M-J, Nelson W, Balbuena TS, Kim R, Kramer R, et al. Next-generation sequencing-based transcriptomic and proteomic analysis of the common reed, Phragmites australis (Poaceae), reveals genes involved in invasiveness and rhizome specificity. American Journal of Botany. 2012;99: 232–247. doi:10.3732/ajb.1100429

24. Jin J, Tian F, Yang D-C, Meng Y-Q, Kong L, Luo J, et al. PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. Nucleic Acids Research. 2017;45: D1040–D1045. doi:10.1093/nar/gkw982

25. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the United States of America. 1992;89: 10915–10919.

26. Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, et al. InterPro: the integrative protein signature database. Nucleic acids research. 2009;37: D211–D215. doi:10.1093/nar/gkn785

27. Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic acids research. 2000;28: 27–30.

28. Bairoch A. The ENZYME database in 2000. Nucleic acids research. 2000;28: 304–305.

29. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, et al. The Pfam protein families database: towards a more sustainable future. Nucleic Acids Research. 2016;44: D279–D285. doi:10.1093/nar/gkv1344

30. Rhee SY, Wood V, Dolinski K, Draghici S. Use and misuse of the gene ontology annotations. Nat Rev Genet. 2008;9: 509–515. doi:10.1038/nrg2363

31. Haas BJ, Papanicolaou A, Yassour M, Grabherr M, Blood PD, Bowden J, et al. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nature protocols. 2013;8: 1494–1512. doi:10.1038/nprot.2013.084

32. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26. doi:10.1093/bioinformatics/btp616

33. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome biology. 2014;15: 550–550. doi:10.1186/s13059-014-0550-8

34. Young MD, Wakefield MJ, Smyth GK, Oshlack A. Gene ontology analysis for RNA-seq: accounting for selection bias. Genome Biology. 2010;11: R14. doi:10.1186/gb-2010-11-2-r14

35. Gardiner-Garden M, Frommer M. CpG islands in vertebrate genomes. J Mol Biol. 1987;196: 261–282.

36. Moreno-Hagelsieb G, Latimer K. Choosing BLAST options for better detection of orthologs as reciprocal best hits. Bioinformatics. 2008;24: 319–324. doi:10.1093/bioinformatics/btm585

37. Valdar WSJ. Scoring residue conservation. Proteins: Structure, Function, and Bioinformatics. 2002;48: 227–241. doi:10.1002/prot.10146

38. Yang Z, Nielsen R. Estimating Synonymous and Nonsynonymous Substitution Rates Under Realistic Evolutionary Models. Molecular Biology and Evolution. 2000;17: 32–43.

39. Lehmann J, Libchaber A. Degeneracy of the genetic code and stability of the base pair at the second position of the anticodon. RNA. 2008;14: 1264–1269. doi:10.1261/rna.1029808

40. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research. 2004;32: 1792–1797. doi:10.1093/nar/gkh340

41. Supek F, Bošnjak M, Škunca N, Šmuc T. REVIGO Summarizes and Visualizes Long Lists of Gene Ontology Terms. PLOS ONE. 2011;6: e21800. doi:10.1371/journal.pone.0021800

42. Zhou A, Yang H, Cui H, Xu H, Ye J, Wang J, et al. WEGO 2.0: a web tool for analyzing and plotting GO annotations, 2018 update. Nucleic Acids Research. 2018;46: W71–W75. doi:10.1093/nar/gky400

43. Soderlund CA, Nelson WM, Goff SA. Allele Workbench: Transcriptome Pipeline and Interactive Graphics for Allele-Specific Expression. PLOS ONE. 2014;9: e115740. doi:10.1371/journal.pone.0115740

44. Cock PJA, Chilton JM, Grüning B, Johnson JE, Soranzo N. NCBI BLAST+ integrated into Galaxy. GigaScience. 2015;4: s13742-015-0080–7. doi:10.1186/s13742-015-0080-7

45. Koski LB, Golding GB. The Closest BLAST Hit Is Often Not the Nearest Neighbor. Journal of Molecular Evolution. 2001;52: 540–542. doi:10.1007/s002390010184

46. Dalquen DA, Dessimoz C. Bidirectional Best Hits Miss Many Orthologs in Duplication-Rich Clades such as Plants and Animals. Genome Biology and Evolution. 2013;5: 1800–1806. doi:10.1093/gbe/evt132