Johnson and Krishnan *Genome Biology* (2022) 23:1
https://doi.org/10.1186/s13059-021-02568-9

Genome Biology

RESEARCH Open Access

Robust normalization and transformation techniques for constructing gene coexpression networks from RNA-seq data

Kayla A. Johnson[1,2] and Arjun Krishnan[1,2*]

* Correspondence: arjun@msu.edu
[1]Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
[2]Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, MI 48824, USA

Abstract

Background: Constructing gene coexpression networks is a powerful approach for analyzing high-throughput gene expression data towards module identification, gene function prediction, and disease-gene prioritization. While optimal workflows for constructing coexpression networks, including good choices for data pre-processing, normalization, and network transformation, have been developed for microarray-based expression data, such well-tested choices do not exist for RNA-seq data. Almost all studies that compare data processing and normalization methods for RNA-seq focus on the end goal of determining differential gene expression.

Results: Here, we present a comprehensive benchmarking and analysis of 36 different workflows, each with a unique set of normalization and network transformation methods, for constructing coexpression networks from RNA-seq datasets. We test these workflows on both large, homogenous datasets and small, heterogeneous datasets from various labs. We analyze the workflows in terms of aggregate performance, individual method choices, and the impact of multiple dataset experimental factors. Our results demonstrate that between-sample normalization has the biggest impact, with counts adjusted by size factors producing networks that most accurately recapitulate known tissue-naive and tissue-aware gene functional relationships.

Conclusions: Based on this work, we provide concrete recommendations on robust procedures for building an accurate coexpression network from an RNA-seq dataset. In addition, researchers can examine all the results in great detail at https://krishnanlab.github.io/RNAseq_coexpression to make appropriate choices for coexpression analysis based on the experimental factors of their RNA-seq dataset.

Keywords: Gene expression, Data normalization, Network reconstruction

Background

Constructing gene coexpression networks is a powerful and widely used approach for analyzing high-throughput gene expression data from microarray and RNA-seq technologies [1]. Coexpression networks provide a framework for summarizing multiple transcriptomes of a particular species, tissue, or condition as a graph where each node

is a gene and each edge between a pair of genes represents the similarity of their patterns of expression. Coexpressed genes are highly likely to be transcriptionally co-regulated and are often functionally related to each other by virtue of taking part in the same biological process or physiological trait [2–5]. Many studies have leveraged these properties to use coexpression networks in several important applications such as determining co-regulated gene groups [6] and associating genes to functions and phenotypes [7].

Nevertheless, multiple experimental factors impact the quantification of the expression of individual genes and the coexpression between pairs of genes, making it necessary to normalize and transform high-throughput gene expression data before downstream analysis. For RNA-seq data, examples of factors that affect the number of reads mapped to a gene include gene length, gene sequence, sample RNA population, and sequencing depth. Some factors have a greater effect on comparisons of gene counts within a single sample ("within-sample" effects) while others have a greater effect on comparisons of the same gene's counts in different samples ("between-sample" effects) [8]. Many data normalization and transformation techniques have been developed to explicitly address one or more of these factors. An additional adjustment that can be considered particularly in coexpression analysis is network transformation, which is applied after calculating correlation between all gene pairs. Coexpression networks are noisy and can indiscriminately capture indirect interactions due to being estimated from noisy, steady-state gene expression data. Hence, previous studies have proposed methods to modify the raw coexpression network to upweight connections that are more likely to be real and downweight spurious correlations based on the topology of the network [9, 10]. Together, appropriately normalizing and transforming RNA-seq data along with adequately transforming the coexpression strengths should yield more accurate estimates of gene-gene coexpression that best capture functional relationships between genes.

However, the best practices for normalization when building a coexpression network from a raw gene-expression dataset have been developed and compared only for data from microarrays [11, 12]. Over the past decade, coexpression network analysis is being routinely applied to the exponentially increasing amount of data from RNA-seq, even though the optimal procedure for network building has not been evaluated and honed for RNA-seq data, particularly in regard to normalization and transformation. Although many normalization strategies have been developed for RNA-seq data, they have mostly been benchmarked only in the context of estimating differential gene expression [13–17]. Very little work has been done so far to comprehensively compare these strategies for normalization and network transformation (and their combinations) to construct the most accurate coexpression networks from RNA-seq data, especially to ensure their robust application to datasets typically generated by individual research groups [1].

The most relevant prior work focuses on establishing best practices that reduce the introduction of artifacts in coexpression networks built from RNA-seq data [18]. This study includes a sequential comparison of a select number of methods for transcript assembly, normalization, and network reconstruction. However, the normalization comparison is based on 10 RNA-seq datasets, leaving considerable room for improvement. First is to increase the number and diversity of datasets studied. This is vital for finding robust procedures that work across datasets that can vary considerably in many

respects, including sample size, sample variability, sequencing depth, tissue type, and other experimental factors. Further, testing on a wide range of datasets is critical both for the analysis of individual datasets as well as integrative analysis of hundreds/thousands of datasets. Second, not only do more normalization and network transformation methods need to be compared but how they might interact in combinations needs to be studied. Third, the resulting networks need to be evaluated directly on the accuracy of the coexpression between gene pairs, instead of performance in a downstream task such as gene function prediction, to ensure maximal utility of the network regardless of the subsequent biological application. Finally, the evaluation metric needs to be informative considering the fact that only a small fraction of all gene pairs in the genome are functionally related.

In this work, we present the most comprehensive benchmarking of commonly used within- and between-sample normalization strategies and network transformation methods for constructing accurate coexpression networks from human RNA-seq data. We tested every possible combination of methods from different normalization and network transformation stages. Our primary interest is in identifying robust combinations of methods that consistently result in coexpression networks that accurately capture general and tissue-aware gene relationships across a large variety of datasets. This will allow us to propose general recommendations useful for experimental research groups analyzing their own RNA-seq data as well as computational researchers seeking to build many coexpression networks from publicly available data for the purposes of data/network integration. Towards this aim, we use hundreds of datasets, generated by a consortium and by individual laboratories, covering multiple experimental factors. We then test the resulting networks on both tissue-naive and tissue-aware prior knowledge about gene functional relationships. Based on these extensive analyses, we finally provide concrete recommendations for normalization and network transformation choices in RNA-seq coexpression analysis.

Results

Expression data, gold standard, and benchmarking summary

To test various within-sample normalization, between-sample normalization, and network transformation methods (and their combinations) on a large data collection, we started with gene count data from the recount2 database [19]. Recount2 contains data from both the Genotype-Tissue Expression (GTEx) project [20] and the Sequence Read Archive (SRA) [21] repository that have been uniformly quality-controlled, aligned, and quantified to the number of reads per gene in the genome. Datasets from the GTEx project allowed us to assess method performance on large, relatively homogeneous datasets with high-sequencing depth and quality. The GTEx data was also critical for investigating the impact of experimental factors such as sample size, which we performed by doing multiple rounds of random sampling from GTEx datasets. Datasets from SRA, on the other hand, were representative of heterogeneous, mostly small experiments (median of 12 samples) that are generated by individual labs, with a range of sequencing depths and qualities. In total, we used 9657 GTEx samples and 6301 SRA samples from a total of 287 datasets (Table 1, Additional file 1: Fig. S1; see the "Methods" section) and processed and evaluated these two collections separately.

Table 1 Summary of data used in this study. See *Figure S1* and the "Methods" section for more details

Data Source	GTEx	SRA
Number of samples	9657 samples	6301 samples
Number of datasets	31 datasets	256 datasets
Number of tissues	31 tissues	19 tissues
Median dataset size	197 samples	12 samples
Total	15,958 samples from 37 unique tissues	

After preprocessing each dataset using lenient filters in order to keep data for as many genes and samples as possible (see the "Methods"), we compared methods commonly used in RNA-seq analysis to effectively construct one coexpression network per dataset (i.e., building 31 GTEx networks and 256 SRA networks). We focused on three key stages of data processing and network building: (a) within-sample normalization: counts per million (CPM), transcripts per million (TPM), and reads per kilobase per million (RPKM); (b) between-sample normalization: quantile (QNT), trimmed mean of *M* values (TMM), and upper quartile (UQ); in addition, we tested two new variations of TMM and UQ—counts adjusted with TMM factors (CTF); counts adjusted with upper quartile factors (CUF)—that directly adjust counts by the size factors but does not correct by library size; and (c) network transformation: weighted topological overlap (WTO) and context likelihood of relatedness (CLR). To systematically examine these methods and their interactions, we built 36 different workflows covering all possible combinations of choices (Fig. 1). For clarity, in the rest of the manuscript, we present individual methods in regular font (e.g., TPM normalization) and italicize workflows (e.g., *TPM*, which is TPM combined with no between-sample normalization and no network transformation, or *TPM_CLR*, which is TPM paired with just CLR). The *Counts* workflow uses no within-sample normalization, between-sample normalization, or network transformation, but is still transformed with the hyperbolic arcsine function.

Since this entire workflow is unsupervised, i.e., not reliant on prior knowledge about gene relationships, we evaluated the resulting coexpression networks by comparing them to gold standards of known gene functional relationships. The gold standards were built using experimentally verified co-annotations to specific biological process terms in the Gene Ontology [22]. These comparisons yielded evaluation metrics that summarize how well the patterns of coexpression captured in the network reflect known gene functional relationships (see *Network Evaluation* in the "Methods" section and *Supplemental Note*). Further, gene activities and interactions vary drastically depending on cell type or tissue. Hence, we also created tissue-aware gold standards to assess whether the resulting networks were able to recapitulate tissue-aware coexpression in addition to general "tissue-naive" coexpression. Tissue-aware gold standards were created for as many tissues as possible by subsetting the naive gold standard using genes known to be expressed in a particular tissue. While area under the receiver operator curve (auROC) is frequently used to estimate network accuracy, it does not account for the fact that only a small fraction of gene pairs (out of the total possible) biologically interact. In the

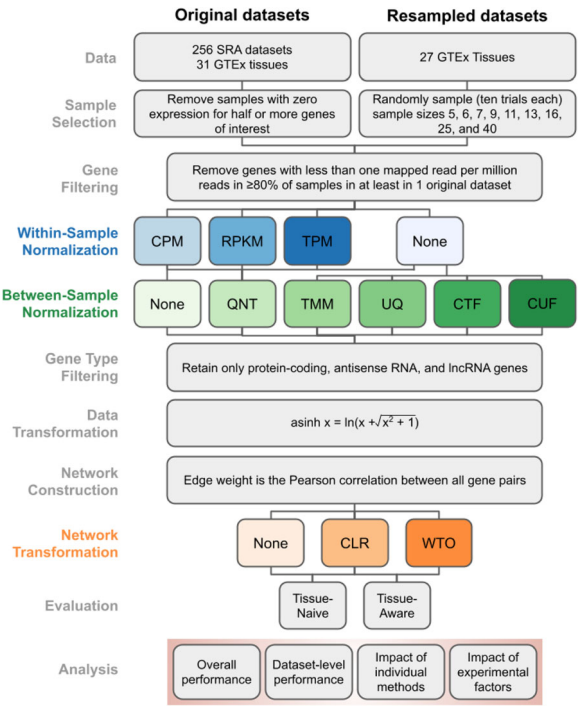


Fig. 1 Pipeline for benchmarking the optimal workflow for constructing coexpression networks from RNA-seq data. The main pipeline was executed for the original GTEx and SRA datasets and a large collection of datasets of different sizes resampled from the GTEx datasets. Three key stages—within-sample normalization, between-sample normalization, and network transformation—where we tested method choices are highlighted in different colors. All the other stages were composed of standard selection, filtering, and data transformation operations. The coexpression networks resulting from all the workflows were evaluated using two gold-standards that capture generic (tissue-naive) and tissue-aware gene functional relationships. Finally, all the evaluation results were used to analyze the impact of various aspects of the workflows, methods, and datasets on the accuracy of coexpression networks. Abbreviations: CPM (counts per million), RPKM (reads per kilobase million), TPM (transcripts per million), QNT (quantile), TMM (trimmed mean of *M* values), UQ (upper quartile), CTF (counts adjusted with TMM factors), CUF (counts adjusted with upper quartile factors), CLR (context likelihood of relatedness), and WTO (weighted topological overlap)

gold standard, this imbalance is reflected by the number of negatives (non-interactions) far outnumbering the positives (interactions) [23]. Therefore, we measured network accuracy using area under the precision recall curve (auPRC), which emphasizes the accuracy of top-ranked coexpression gene pairs [24].

In total, for each of the 287 datasets from GTEx and SRA, we built one coexpression network per dataset using each of the 36 workflows, resulting in 8610 coexpression networks. Later on, we created 2430 additional datasets generated by resampling GTEx that, when run through all the workflows, resulted in another 72,900 networks. Each GTEx network contains 20,418 genes while each SRA network contains 22,084 genes, and all networks are fully connected with edges weighted by their strength of correlation. Each of these networks were evaluated using the tissue-naive gold-standard and, whenever applicable, the tissue-aware gold-standard. Finally, we replicated the analysis of the top-performing workflows using as many matched SRA datasets as possible from another RNA-Seq repository, refine.bio [25], where read alignment and expression quantification were done using different methods than recount2.

Overall performance of workflows

For all 36 workflows, Fig. 2 shows the overall performance of the networks resulting from GTEx (left) and SRA (right) recount2 datasets based on evaluation using the tissue-naive gold standard. Figure S2 shows the performance of these networks based on the tissue-aware gold standards (when available). Overall, networks built from GTEx datasets are far more accurate than those built from SRA datasets (Fig. 2, S2). In each of the four cases—GTEx and SRA networks evaluated using tissue-naive and tissue-aware gold standards—most of the top-performing workflows contain CTF or CUF normalization. Further transforming the network with CLR (*CTF_CLR* and *CUF_CLR*) results in top-tier workflows for the GTEx datasets regardless of gold standard. However, CLR transformation is only among top-performing methods for SRA datasets in recovering tissue-aware gene relationships. Though *CTF_CLR* and *CUF_CLR* still perform quite well on the tissue-naive standard for SRA, there is a clear gap from the top tier. Despite CTF- and CUF-containing workflows resulting in top performances, workflows that include other between-sample normalization methods are absent among the top ten workflows for both GTEx and SRA. Workflows with TMM or UQ seem to be more comparable to workflows using within-sample normalization methods.

The next noteworthy observation is that the top-tier workflows do not include a within-sample normalization step. Yet, workflows that do include within-sample normalization methods (CPM, RPKM, TPM) can perform better than many other

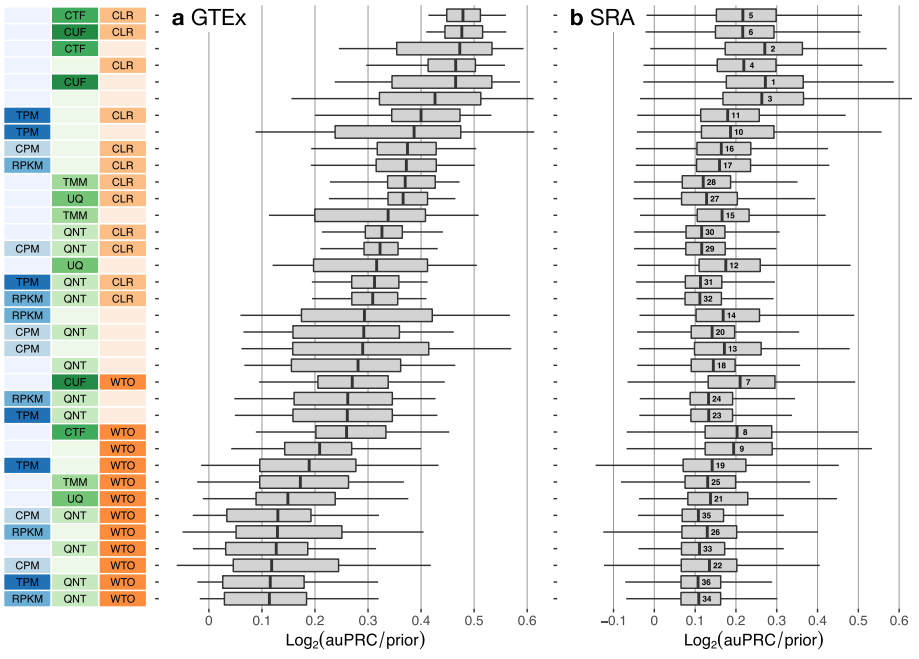
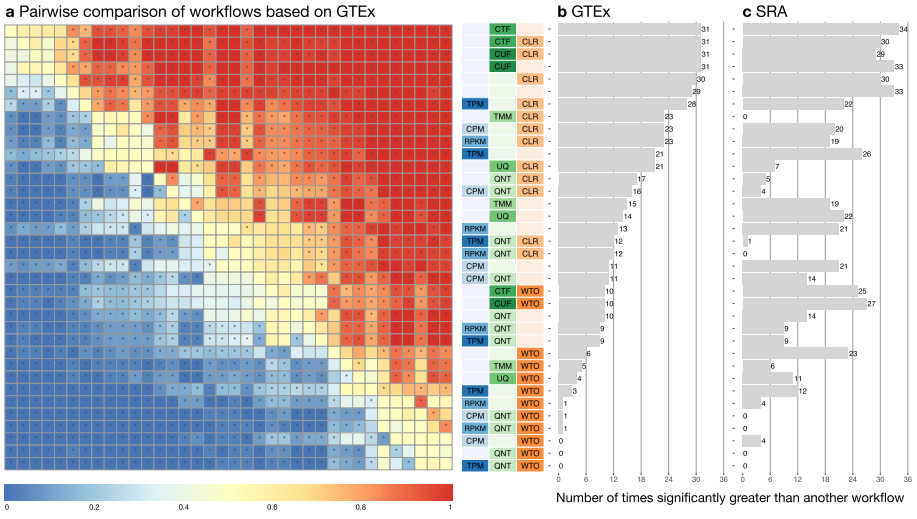


Fig. 2 Overall performance of workflows. The plots show the aggregate accuracy of all coexpression networks resulting from each individual workflow using **a** GTEx and **b** SRA datasets, evaluated based on the tissue-naive gold standard. The workflows (rows) are described in terms of the specific method used in the within-sample normalization (blues), between-sample normalization (greens), and network transformation (oranges) stages. The performance of each workflow is presented as boxplots (without outliers) that summarizes the log2(auPRC/prior) of each workflow where auPRC is the area under the precision recall curve (see the "Methods"). The workflows are ordered by their median log2(auPRC/prior) for the GTEx data. The numbers inside the SRA boxes indicate rank by median log2(auPRC/prior) of the workflows for the SRA data. *Figure S2* contains these plots based on the tissue-aware gold standard

workflows depending on other choices made in the pipeline, the best choice often is to be paired with no other method or CLR alone. For GTEx datasets, CLR seems to generally result in slightly improved performance, while the WTO transformation almost exclusively makes up the bottom tier of workflows. For building networks from SRA datasets, although workflows including WTO do not exclusively end up in the bottom tier (as is the case with GTEx data), adding WTO to a particular workflow always hurts performance. The worst workflows for SRA in either standard are quantile normalization (QNT) paired with CLR or WTO.

Dataset-level performance of workflows

Next, we dissected the aggregated results described above for GTEx and SRA as a whole by examining the accuracy of these workflows on a per-dataset basis. First, we compared pairs of workflows to each other and determined the proportion of datasets in which one workflow outperformed the other across all GTEx and all SRA datasets (Fig. 3, S3–5, heatmap colors). Second, we performed paired statistical tests to estimate



d Most significant workflows in each dataset collection in each evaluation setting, along with the number of times each workflow outperformed any other workflow

GTEx				SRA			
Tissue-naive		Tissue-aware		Tissue-naive		Tissue-aware	
CTF	31	*CTF_CLR*	28	*CTF*	34	*CUF*	33
CTF_CLR	31	*CUF_CLR*	28	*CUF*	33	*CTF*	33
CUF_CLR	31	*CLR*	25	*Counts*	33	*Counts*	33
CUF	31	*TMM_CLR*	11	*CTF_CLR*	30	*CTF_CLR*	31
CLR	30	*TMM*	11	*CLR*	30	*CLR*	31

Fig. 3 Dataset-level pairwise comparison of workflow performance. **a** The heatmap shows the relative performance of a pair of workflows, corresponding to a row and a column, directly compared to each other for the GTEx datasets based on the tissue-naive gold standard. The workflows along the rows are depicted using color swatches similar to Fig. 2. The color in each cell (row, column) represents the proportion of datasets for which the workflow along the row has a higher log2(auPRC/prior) than the workflow along the column. Comparisons that are statistically significant (corrected $p < 0.01$) based on a paired Wilcoxon test are marked with an asterisk. *Figure S3* contains the corresponding heatmap for the SRA datasets. **b**, **c** Barplots show the number of times each workflow was significantly greater than another workflow for GTEx and SRA datasets. *Figures S4* and *S5* contain these performance plots based on the tissue-aware gold standard. **d** The table shows the most significant workflows across evaluation cases along with the number of times a given workflow outperformed any other workflow for the GTEx and SRA datasets based on the tissue-naive and tissue-aware gold standards

the significance of the difference between the workflows (Fig. 3, S3–5, asterisks on the heatmap). Finally, we scored each workflow based on the number of other workflows it significantly outperforms (Fig. 3, S4 barplots). Based on this analysis, in the "GTEx-naive" setting (i.e., networks from GTEx data evaluated on the tissue-naive gold standard), we observed that five workflows are all significantly more accurate than 31 other workflows but not significantly different from one another (paired Wilcoxon rank-sum test; corrected p value < 0.01; Fig. 3). Within these four workflows, *CTF* outperforms *CTF_CLR*, *CUF*, and *CUF_CLR* on 58%, 61%, and 58% of GTEx networks, respectively. The *CTF* workflow is also significantly better the most number of times compared to other workflows in the SRA networks using the naive standard, although *Counts* and *CUF* are only slightly behind *CTF* (Fig. 3, S3). These workflows tie for first place when SRA networks are evaluated on the tissue-aware gold standards (Additional file 1: Fig. S4, S5).

When the GTEx networks are evaluated on tissue-aware standards, there are much fewer significant differences between workflows overall, with the exception of *CTF_CLR*, *CUF_CLR*, and *CLR* being significantly greater than 28, 28, and 24 workflows, respectively (Additional file 1: Fig. S4). Here, *CTF_CLR* performs better than *CUF_CLR* on 57% of networks and better than *CLR* on 76% of networks. Despite having similar median $\log_2$(auPRC/prior) values to *CTF_CLR* and *CUF_CLR* (Additional file 1: Fig. S2), the *CUF* and *CTF* workflows only perform significantly better than another workflow a handful of times (Additional file 1: Fig. S4). This suggests that including CLR in the workflow is especially helpful in capturing tissue-aware coexpression in the GTEx networks.

Again, the impact of within-sample normalization varies depending on the choice of the other methods in the workflow. *TPM_CLR* is generally the top-performing workflow among those including within-sample normalization across evaluation cases, though *TPM* slightly outperforms *TPM_CLR* for the SRA networks evaluated on the naive standard (Fig. 3 and S3).

The impact of network transformation is similar between GTEx and SRA data, but there is disagreement in the very top method. With GTEx, workflows that include CLR tend to be significant the most number of times, while WTO-containing workflows tend to be the least. Not a single workflow with WTO significantly outperformed any workflow without it for GTEx based on the tissue-aware gold standard (Additional file 1: Fig. S4). On the other hand, CLR workflows perform well on the SRA networks, but do not constitute the workflows that were significantly greater than another the absolute most number of times (Additional file 1: Fig. S3 and S5). WTO hurts performance in every case even here. Pairing either CLR or WTO with quantile normalization (QNT) yields particularly poor performance in the SRA networks. All together, these results suggest that CTF yields the most accurate coexpression network by a very close margin and CLR can further improve the network in select cases.

Impact of individual methods on performance of workflows

Though the previous analyses shed light on the contributions of individual methods, we wanted to more explicitly assess how choosing or not choosing a particular within-sample normalization, between-sample normalization, or network transformation

affects general performance of any given workflow. To this end, for each method, we calculated the proportion of times that workflows that include a particular method performed significantly better than workflows that did not include the method (Fig. 4; see the "Methods" for details).

This analysis clearly shows that, in all four cases (GTEx and SRA, each with tissue-naive and tissue-aware standards), utilizing any within-sample normalization method results in worse overall performance than not using it (Fig. 4 and S6). Among within-sample normalization methods, TPM usually performs slightly better than CPM and RPKM. CTF and CUF are the best between-sample normalization methods. Their performances are exactly equal for GTEx data evaluated on either standard and for SRA data evaluated on the naive standard; CTF is slightly better than CUF for SRA data in the tissue-aware standards. However, doing no between-sample normalization performs quite well too, only narrowly worse than CTF or CUF. It is clear in all four cases that TMM, UQ, and quantile normalization (QNT) are vastly outperformed. Network transformation is the group most obviously different between GTEx and SRA data, with CLR being the clear winner for GTEx, while not doing any network transformation is significant many more times for SRA regardless of gold standard (Fig. 4 and S6).

Impact of varying experimental factors on performance of workflows

The reason we included SRA data in this study is that SRA datasets are very representative of expression datasets typically generated by numerous individual laboratories. Accordingly, these datasets vary considerably in terms of multiple factors including sample size, sample similarity, number of mapped reads, and tissue type. Though these factors impact the quality of coexpression networks derived from the individual datasets, it is hard to tease out the effect of each of these factors (controlling for others) on

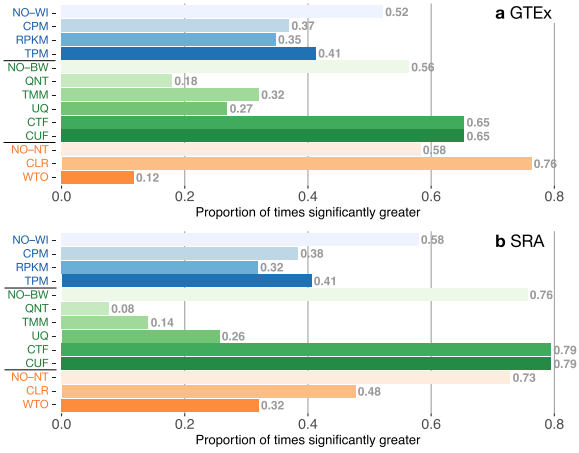


Fig. 4 Impact of individual methods on performance of workflows. Each bar in the two barplots, corresponding to a specific method, shows the proportion of times (*x*-axis) that workflows including that particular method (*y*-axis) were significantly better than other workflows. The barplots correspond to performance for the **a** GTEx and **b** SRA datasets evaluated on the tissue-naive gold standard. In order to make the comparison of between-sample normalization methods fair, workflows also including CPM, RPKM, or TPM were left out because it is not possible to pair them with CTF/TMM/CUF/UQ normalization. Similarly, CTF/TMM/CUF/UQ methods are not included for "no within-sample normalization" (NO–WI). *Figure S6* contains these barplots based on the tissue-aware gold standard

the accuracies that we observed using different workflows on SRA data. Therefore, using the large GTEx datasets, we created a collection of SRA-like datasets to more closely examine the impact of each experimental factor. First, we determined the nine sample sizes (5, 6, 7, 9, 11, 13, 16, 25, and 40) that are representative of SRA datasets. Then, from each GTEx tissue dataset with at least 70 samples, we randomly selected samples to create ten datasets for each sample size (see the "Methods" section). We then applied all 36 workflows to construct coexpression networks from each one of these datasets. The resulting 72,900 networks were used to investigate the effects of varying each experimental factor by counting the number of times a given workflow significantly outperformed any other workflow (Fig. 5). In addition to this analysis with these resampled data, we also examined the effect of sample similarity and number of mapped reads (see Experimental factor analysis in the "Methods" section) directly in the SRA data by splitting the datasets into five equal size bins based on each of these factors and determining the number of times a given workflow was significantly better than another within each bin (Additional file 1: Fig. S7).

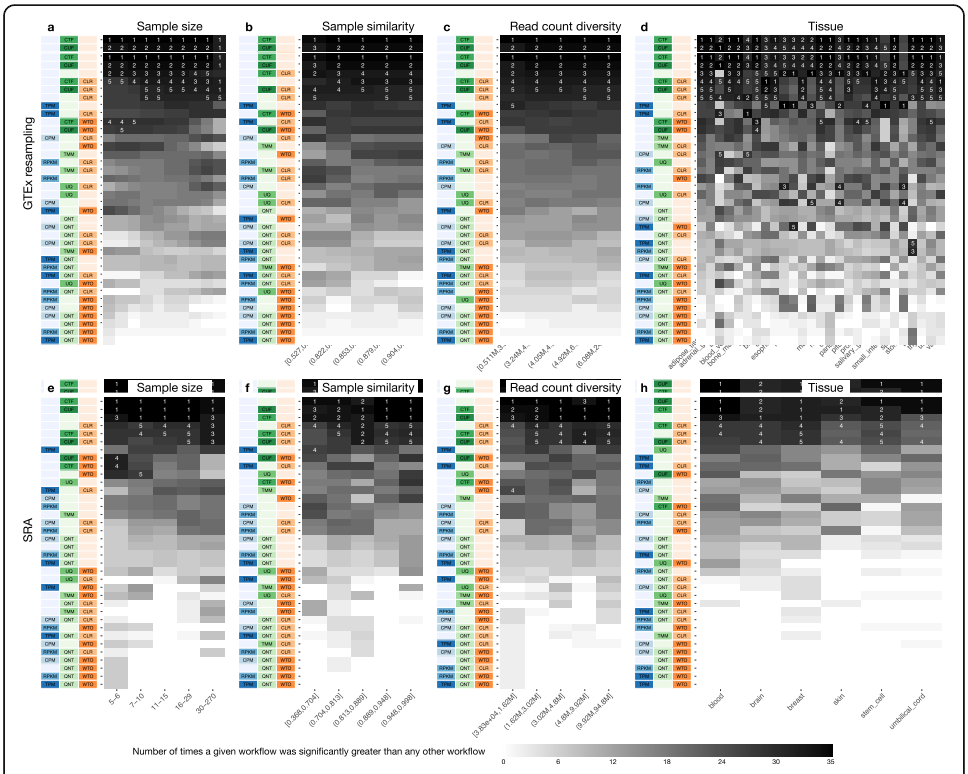


Fig. 5 Impact of various dataset-related experimental factors on performance of workflows. Each heatmap shows the number of times (cell color) each workflow (row) outperforms other workflows as a particular experimental factor pertaining to the input datasets is varied (columns), when the resulting coexpression networks are evaluated based on the tissue-naive gold standard. The darkest colors indicate workflows that are significantly better than the most other workflows. In addition, the top 5 workflows in each column are marked with their rank, with ties given minimum rank. The heatmaps on the top (**a–d**) correspond to datasets from GTEx resampling and those on the bottom (**e–h**) correspond to SRA datasets. The heatmaps from left to right show workflow performance by sample size (**a**, **e**; number of samples used to make the coexpression network), sample similarity (**b**, **f**; median spearman correlation of 50% most variable genes between samples), read count diversity by counts (**c**, **f**; standard deviation of counts sums across samples), and tissue of origin (**d**, **h**). *Figure S7* contains these heatmaps based on the tissue-aware gold standard

In the GTEx-resampled data, *CTF* was significantly better than all other workflows for sample sizes 5 through 40 when using the naive standard for assessment (Fig. 5). *CUF* is a close second, performing significantly better than all workflows other than *CTF* at sample sizes 7 through 40. Using only *Counts* (no normalization) is surprisingly effective, especially at lower sample sizes, while *CTF_CLR* and *CUF_CLR* improve performance with increasing sample size. In fact, when all samples from a given GTEx tissue are used (≥70 samples), there is no significant difference between *CTF*, *CUF*, *CTF_CLR*, and *CUF_CLR*. *CLR* is the next best workflow after those top four. The only other workflows that are ever ranked in the top five are *CTF_WTO* and *CUF_WTO* and that too only at low sample sizes (5–7). Based on the tissue-aware standards, *CTF_CLR* is the most effective workflow on all sample sizes except 5, where *CTF* and *CUF* are the top workflows (Additional file 1: Fig. S7). For the highest two sample sizes (25 and 40), *CTF_CLR* is substantially better than all other workflows. The only workflows ranked in the top five in sample sizes 5 through 40 are *CTF_CLR*, *CUF_CLR*, *CLR*, *CUF, CTF,* and *TPM_CLR*. *CTF* and *CUF* also perform well on the SRA data evaluated on the naive standard, being the top workflows in all five sample size groups (Fig. 5). Performance on the tissue-aware standards is slightly more variable, with *Counts*, *CTF*, and *CUF* being top ranked in lower sample size groups and *CLR*, *CUF_CLR*, and *CTF_CLR* performing better in the highest sample size group (Additional file 1: Fig. S7). Again, it is clear that CTF and CUF are superior methods, with CLR improving performance in select cases.

Sample similarity and read count diversity analyses show similar results to those from sample size analysis. When evaluating the GTEx-resampled data on the naive standard, *CTF* is almost always significantly better than every other workflow across all groups, while evaluating on the tissue-aware gold standards ranks *CTF_CLR* as the top workflow most consistently (Fig. 5, Additional file 1: Fig. S7). In both standards, *CTF*, *CUF*, *CLR*, *CTF_CLR*, *CUF_CLR,* and *Counts* are the workflows consistently showing up in the top five ranks. The SRA networks evaluated on either standard have *CTF*, *CUF*, and *Counts* showing up in the top three ranks across most groups, with *CLR*, *CTF_CLR*, and *CUF_CLR* making up most of the other workflows in the top five ranks (Fig. 5, Additional file 1: Fig. S7).

Tissue is the factor that shows the most variability in terms of what makes up the top workflows, especially when evaluating on tissue-aware gold standards. This is due in part to the fact that splitting experiments by tissue results in the smallest groups, making significance more difficult to detect. Nevertheless, the top workflows from the analyses of other factors still have the best overall performance across all tissues. In the GTEx-resampled data, *CTF* is the top-ranked workflow most often based on the naive gold standard. *CUF* and *Counts* are almost always in the top five most significant workflows, while *CTF_CLR*, *CUF_CLR*, and *CLR* show up often. When evaluated on tissue-aware gold standards, *CTF, CLR, and CTF_CLR* are ranked number one more frequently than any other workflow, but they are not as consistent as *CTF* in the naive standard. *CUF* and *CUF_CLR* are the other top-performing workflows, but a handful of other workflows enter the top five ranks in at least a few tissues. For SRA, only tissues that had more than fifteen separate experiments were used in the significance analysis (Additional file 1: Fig. S1). On the naive standard, *CUF*, *CTF*, or *Counts* were always the most significant workflow in any given tissue and *CLR*, *CTF_CLR*, and *CUF_CLR*

were usually in the top five. A similar if less consistent pattern can be observed from the tissue-aware evaluations. Taken together, these results suggest that the top-performing methods are largely robust to common experimental factors that vary from experiment to experiment. This property is critical because, to be practically beneficial, the best workflow for constructing coexpression networks should result in accurate coexpression networks irrespective of variations in these experimental factors.

Impact of varying alignment and counts quantification performance of workflows

So far, our analysis has considered datasets from the recount2 database. This has allowed us to evaluate the performance of each workflow on a large, diverse set of datasets which have been uniformly aligned and transformed into gene counts. However, this begs the question of whether the observed results—especially the top performance of *CTF*, *CUF*, and *Counts*—would hold when different methods for read alignment and counts quantitation are used. To determine whether this is the case, we matched as many of our recount2 SRA datasets as possible to those from refine.bio [25], another RNA-seq repository that uses completely different methods for alignment and quantification. This turned out to be 186 datasets in the naive evaluation and 163 of those could be evaluated with a tissue-aware standard. Unfortunately, GTEx data is not available from refine.bio. In this new analysis, we left out the worst performing methods in each tested category, i.e., RPKM, QNT, and WTO for within-sample normalization, between-sample normalization, and network transformation, respectively. This leaves us with 14 workflows to evaluate on the refine.bio datasets.

In the naive evaluation, *CTF*, *CUF*, and *Counts* are once again the top-tier workflows. However, the *CUF* workflow significantly outperforms the other two across all datasets (Fig. 6). The second tier consists of *CTF_CLR*, *CUF_CLR*, and *CLR*, though it is not quite as well separated from the remaining workflows. The tissue-aware evaluation shows much less separation between *CUF*, *CTF*, *Counts*, *CTF_CLR*, *CUF_CLR*, and *CLR* in terms of overall performance measured by $\log_2$(auPRC/prior), but *CTF* and *CUF* significantly outperform more workflows than any other (Additional file 1: Fig. S8). In summary, we replicated the ranking of coexpression workflows using RNA-seq data processed with an entirely different pipeline for alignment and quantification.

The general trends presented above are all based on network accuracy measured using a metric based on the area under the precision-recall curve ($\log_2$(auPRC/prior)). These trends also hold when network accuracy is measured using precision at low recall, which focuses on maximizing the number of functional gene pairs among the high-scoring gene pairs instead of focusing on recovering all functional gene pairs. Put another way, the trends described above hold even when a threshold is applied to the coexpression network to retain just the high-scoring gene pairs for subsequent analysis. For the sake of completion, we have also evaluated all networks using the area under the ROC curve (auROC). All these results based on three different evaluation metrics ($\log_2$(auPRC/prior), precision at 20% recall, and auROC) are available as a consolidated webpage at https://krishnanlab.github.io/RNAseq_coexpression that researchers can explore to easily examine the performance of various workflows based on the properties of their RNA-seq dataset.

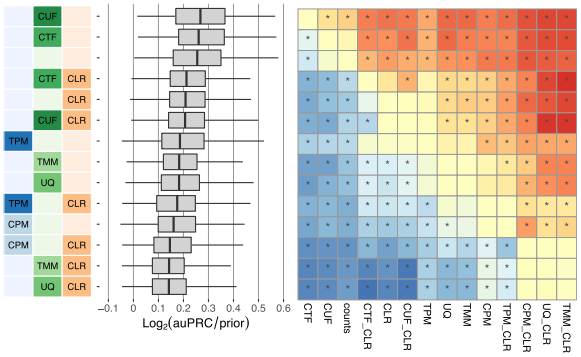


Fig. 6 Overall performance of workflows and pairwise-comparison using refine.bio datasets. The boxplots show the aggregate accuracy of all coexpression networks resulting from each individual workflow using SRA datasets in refine.bio, evaluated based on the tissue-naive gold standard. The performance of each workflow is presented as boxplots (without outliers) that summarizes the log2(auPRC/prior) of each workflow, where auPRC is the area under the precision recall curve (see the "Methods"). The workflows are ordered by their median log2(auPRC/prior). The heatmap shows the relative performance of pairs of workflows (rows and columns) compared to each other for the refine.bio SRA datasets based on the tissue-naive gold standard. The color in each cell (row, column) represents the proportion of datasets for which the workflow along the row has a higher log2(auPRC/prior) than the workflow along the column. Comparisons that are statistically significant (corrected $p < 0.01$) based on a paired Wilcoxon test are marked with an asterisk. *Figure S8* contains these plots based on the tissue-aware gold standard

Discussion

Despite the utility and growing popularity of coexpression analysis of RNA-seq data, relatively little focus has been devoted to identifying the optimal data normalization and network transformation methods that result in accurate RNA-seq-based coexpression networks. Here, we present the most comprehensive analysis of the effects of commonly used techniques for RNA-seq data normalizations and network transformation on gene coexpression network accuracy (Fig. 1). We implemented 36 network-building workflows—one for every combination of within-sample normalization, between-sample normalization, and network transformation methods—and we ran each workflow on hundreds of RNA-seq datasets from GTEx and SRA. The resulting coexpression networks were evaluated using both known tissue-naive and tissue-aware gene functional relationships to ensure that the networks were tested for capturing not just generic gene interactions but also interactions relevant to the tissue under consideration (Additional file 1: Fig. S9, Fig. S10). The evaluations shed light on several key aspects of the impact of within-sample normalization, between-sample normalization, and network transformation methods (and their interplay) on the accuracy of the resulting coexpression networks.

Impact of within-sample normalization

Within-sample normalization—commonly executed by converting gene counts to CPM, RPKM, or TPM—corrects for factors such as library size and gene length. As gene length biases both gene counts and their downstream analysis [26], it is not very surprising that TPM usually outperforms CPM, as CPM only corrects for library size and not gene length. However, the order in which gene-length and library-size correction are combined appears to be important. For example, studies have shown that

RPKM, which first corrects for library size and then for gene length, is inferior to other methods in differential expression analysis and is not recommended [13–15]. Some studies have also noted that using RPKM does not necessarily take away the length bias in gene expression and can be unduly influenced by relatively few transcripts [13, 27]. TPM was proposed as an improvement over RPKM by first correcting for length and then by library size. Thus, the resulting expression values more accurately reflect the "relative molar concentration" of an RNA transcript in the sample [28]. TPM normalization scales every sample to the same total RNA abundance (i.e., the same total sum of TPM values). Thus, gene expression across samples becomes more comparable when TPM normalized than when RPKM normalized. Consistent with these previous studies, we find that RPKM generally results in lower-performing coexpression networks and that TPM consistently outperforms CPM and RPKM, and can even occasionally perform better than the general top-performers CTF and CUF. Finally, since a number of technical and biological factors affect the size and makeup of the sample library, TPM has been found to be most effective when comparing samples from the same tissue type and experiment [29]. This observation could explain the good performance of TPM in our work wherein only samples within a dataset are compared and analyzed together to construct a coexpression network.

Impact of between-sample normalization

Next, our results reinforce the expectation that between-sample normalization (using techniques such as CTF and CUF) leads to the largest improvement in coexpression accuracy. These methods are designed to make expression values across samples more comparable to one another, an aspect critical for coexpression analysis. However, QNT, a between-sample normalization method that is most commonly used with microarray data, performs very poorly for RNA-seq data. This is likely because QNT forces the distribution of samples to be exactly the same, meaning that each gene value is forced to be a particular quantile value. Consequently, it does not suit situations where there truly are different numbers of genes that are expressed outside of the typical ranges across samples [8, 30], an effect that is further exacerbated in RNA-seq data because it has a larger dynamic range than microarray data. Genes with extreme values would not influence CTF or CUF normalization because they are explicitly excluded from the calculation of adjustment factors. CTF specifically finds a subset of genes that are probably not differentially expressed between samples to make gene values comparable across the entire group, while CUF uses only the upper quartile gene values to adjust samples. This makes both normalizations robust to a number of highly or lowly expressed genes. However, large-scale changes in gene expression or high amounts of asymmetry, e.g., a large difference in the number of genes expressed above the typical range versus expressed below the typical range, violate these assumptions [8]. In our test cases, CTF and CUF performed the best, but it is possible that violation of their base assumptions may occur in specific disease conditions or external perturbations, leading to a significant decrease in their performances. The relatively lower performance of TMM and UQ, which are essentially CTF and CUF with library size correction, implies that library size correction is not the most helpful normalization strategy for building coexpression networks based on linear correlation measures. As noted below, measures such as

Pearson correlation automatically include a standardization of gene expression across samples, which could explain why additional library size correction may not be needed. This implication is also supported by the *Counts* workflow outperforming within-sample normalization workflows.

Impact of network transformation

Network transformation is where there is most disagreement between GTEx and SRA data. CLR was the best network transformation method for GTEx data, while doing no transformation of the coexpression values gave the best results for SRA data. The most pronounced factor that explains this difference is sample size. The median sample size of SRA datasets is 12, while that of GTEx datasets is 197. Only four GTEx datasets have less than 70 samples (Additional file 1: Fig. S1). Furthermore, GTEx resampling analysis showed that *CTF_CLR* and *CUF_CLR* improve with increasing sample size on the naive standard (Fig. 5) and to a lesser extent on the tissue-aware standards (Additional file 1: Fig. S7) since CLR tended to already have better performance in general on tissue-aware standards than on the naive gold standard. For each gene pair, CLR adjusts the edge weight based on its value in relation to the distribution of edge weights for the individual genes in that pair to all other genes in the network. So, our hypothesis is that having a larger sample size results in a better estimate of each edge weight as well as the distribution of edge weights for each gene, which in turn increases CLR's accuracy. Supporting this hypothesis, other studies have noted an association between larger sample size and more accurate coexpression networks [18, 27] and subsequent network transformation with CLR [31]. WTO, on the other hand, performs poorly for both GTEx and SRA data. WTO adjusts the edge weight between gene pairs based on whether they share strong connections to the same set of genes in the network. Therefore, while CLR relies on summary statistics (mean and standard deviation) of edge distributions to adjust the edge weight between each gene pair, WTO relies on the actual, likely noisy, coexpression values, which may contribute to its inferior performance. It is also possible that CLR's strategy more effectively deals with the mean-correlation relationship bias, or the observation that highly expressed genes tend to be more highly coexpressed, by capturing them as summary statistics, without relying on the fact that each of the correlation estimates are correct [32, 33]. This may, in turn, explain why CLR tends to perform better on tissue-aware gold standards than on our naive gold standards, since genes that are ubiquitously expressed (and therefore involved in general, tissue-naive interactions) tend to be more highly expressed [34].

Impact of data transformation

RNA-seq data analyses typically benefit from a data transformation that stabilizes the variance across mean values, i.e., renders the data more homoskedastic, because, in its untransformed form, the expected variance grows with the mean for gene counts [35]. A standard procedure when working with RNA-seq (or even microarray) data for either differential expression analysis or coexpression analysis is to log transform gene counts. Since gene counts for several genes can be zero, the typical manner in which log transformation is applied to RNA-seq data is to add a pseudocount (of 1, for example) to every gene's count (say, "x") before taking the log (i.e., $\log(x + 1)$).

However, adding a constant pseudocount to all genes is disadvantageous because low gene counts are disproportionately increased compared to high gene counts before log transformation (e.g., 1 + 1 is a 100% increase for a gene count of 1, but 941 + 1 is almost a negligible increase). The hyperbolic arcsine (asinh) transformation—$\log(x + \sqrt{(x^2 + 1)})$—mitigates this effect [36]. The asinh function is defined along the entire real number line and circumvents the need for predefining a constant pseudocount and instead calculates a pseudocount for each gene that is proportional to that gene's original count. Therefore, it has a compression effect like the natural log function but much less so for small values of x [37]. Due to this advantage, each of our workflows uses the asinh transformation. However, since asinh has not been explicitly tested before (to the best of our knowledge), we analyzed the impact of this transformation on the coexpression network accuracy. We find that the asinh transformation yields an improvement in performance over no data transformation for our top ten workflows in GTEx and SRA datasets (Additional file 1: Fig. S11). It is worth noting that the *Counts* workflow performs well despite not incorporating any within- or between-sample normalization but only an asinh transformation. We speculate that this good performance is due to the variance stabilization provided by the asinh transformation along with the across-sample normalization of gene expression vectors inherent within the calculation of the Pearson correlation coefficient.

The popular R package for differential expression analysis, DESeq2 [35], offers two other data transformations for gene counts: variance stabilizing transformation (VST) [38] and regularized logarithm transformation (rlog) [35]. Both transformations are similar to the log transformation of adjusted counts along with a pseudocount parameter that is chosen in a data-driven manner. These transformations consider between-sample effects like library size and are designed to only be used on counts data as part of calculating differential gene expression. Nevertheless, these transformations could in theory be applied to coexpression analysis. Hence, we compared asinh, VST, and rlog along with their combinations with network transformation methods and found that asinh is the best transformation for coexpression analysis in our all evaluations (Additional file 1: Fig. S12–15). The VST and rlog may perform better when supplied with sample group information. Therefore, we do not recommend the use of either transformation in DESeq2 for large-scale application to publicly available RNA-seq datasets for coexpression analysis.

Recommendations for building coexpression networks from RNA-seq data

By constructing coexpression networks for diverse datasets from both GTEx and SRA, we were able to evaluate workflows on large, homogeneous datasets as well as smaller, heterogeneous datasets to identify methods that are robust to differing technical and biological factors. Although there is some variation in performance between GTEx and SRA data, and slightly more variation introduced by tissue-aware gold standards, many trends are consistent across datasets and evaluations. Based on all our results, we make the following recommendations for building coexpression networks from RNA-seq data using Pearson or Spearman correlation:

- If gene counts are available, use CTF or CUF to normalize the data. They consistently give the best performance regardless of various factors. Between the two, CTF seems to be slightly more consistent in yielding top performance. Even though no normalization (*Counts*) leads to good performance in our study, applying the additional normalization step is prudent to ensure robustness against variabilities specific to a new dataset.
- If data is only available after within-sample normalization, use TPM for coexpression analysis. Data in CPM and RPKM units can be easily converted to TPM. TPM outperforms CPM and RPKM and yields consistently reasonable performance.
- After normalization, perform log transformation (using asinh) and calculate coexpression using Pearson correlation coefficient.
- If the dataset has greater than 40 samples, use CLR to transform the pairwise gene correlations. CLR may also help certain cases where the main interest is interactions that are specific to a given tissue.
- QNT and WTO hurt performance in combination with every other method, in all cases, and should not be used.

To enable researchers to explore all our analyses in a streamlined manner and find the results most relevant to their own RNA-seq datasets of interest, we have made them available as a rich webpage written with R Markdown: https://krishnanlab.github.io/RNAseq_coexpression.

Potential future applications and extensions

Going forward, we can leverage this comprehensive benchmarking framework for coexpression analysis to answer newer and subtler questions about data quality and sample composition. For example, many studies have found that removing unwanted variation, i.e., noise caused by technical rather than biological factors, in the RNA-seq data has led to improvements in downstream analysis including the calculation of coexpression networks [39, 40]. Such corrections are often done using SVD-based methods, including removing the first (or the first few) principal components. However, caution must be taken when using these methods as they may easily remove biological signals from the data [41], especially in typical small-to-medium-sized datasets produced by most research labs (e.g., represented in SRA). Future work using our framework could help learn the guidelines for deciding which and how many factors to remove while carefully considering the various properties of the data and the biological objective of the analysis. For instance, one could explore if different tissues might be sensitive to different technical factors; signal from blood is often heavily influenced by the large variation in cell type composition but the brain is much more greatly affected by the post-mortem-interval [42]. Another related and open question is how cell type composition influences gene coexpression calculated from bulk tissue data. Some studies have concluded that gene coexpression networks are heavily confounded by this factor [43, 44], while others have shown that coexpression derived from single-cell data is very similar to bulk coexpression [45, 46]. Finally, a similar framework could also be used to explore the best procedure for building coexpression networks from single-cell RNA-seq data,

which has an entirely different set of challenges [47] that call for an entirely separate benchmarking effort.

Conclusions

We have performed an extensive benchmarking and analysis of how data normalization and network transformation impact the accuracy of coexpression networks built from RNA-seq datasets. Based on this work, we have arrived at concrete recommendations on robust procedures that will generally lead to best coexpression networks. Specifically, using Counts adjustment with TMM Factors (CTF) and Counts adjustment with Upper quartile Factors (CUF) normalizations to construct coexpression networks results in the most consistently high accuracy networks, and using CLR to transform the network can further increase accuracy in select cases. All the results from this study—for GTEx, SRA, and GTEx resampling datasets, based on tissue-naive and tissue-aware gold standard, using three different evaluation metrics—are available as a consolidated webpage at https://krishnanlab.github.io/RNAseq_coexpression. Researchers can use this website to easily examine the performance of various workflows and make appropriate choices for coexpression analysis based on the properties of their RNA-seq dataset of interest. All the scripts to reproduce our results are available at https://github.com/krishnanlab/RNAseq_coexpression [48], along with scripts that researchers can use to create coexpression networks from their datasets of interest. Finally, all the coexpression networks constructed in this study are available at https://doi.org/10.5281/zenodo.5510567 [49].

Methods

Data collection

Read counts for both SRA and GTEx datasets were downloaded from the recount2 database [19] and processed separately. Recount2 aligns all sequenced reads using Rail-RNA, which eliminates the effect of using different alignment software on separate experiments. We obtained SRA data for any tissue with at least five separate experiments that each had at least five samples. The set of samples from each experiment (project) was considered as an individual dataset from which coexpression networks are inferred (one network per dataset). If a given experiment had samples from multiple tissues, the samples were divided into multiple datasets that each contain samples from the same tissue to yield 543 candidate SRA datasets. We downloaded all available GTEx data, which was a total of 9657 samples from 31 tissues.

Preprocessing

As a form of quality control, we excluded experiments that recount2 identified as having a misreported paired-end status. Experiments that contained "cell line," "celll line," "passage," "cultured cells," or "cell culture" in the characteristics metadata were also removed so as to retain primary tissue samples, which left 341 SRA datasets. Next, we discarded low-coverage samples that had zero expression (counts) in at least half of all genes of interest (lncRNA, antisense RNA, and protein-coding genes) and subsequently excluded entire datasets that no longer contained five or more samples. Any dataset that had a sample removed under these criteria was not retained due to dropping below

five samples. Retaining only tissues that still had at least 5 separate experiments left 256 datasets. Finally, we removed genes with very low expression across the board by filtering out those that did not have at least one read per million sample reads in at least 20% of the samples in at least one dataset. This resulted in 22,084 genes in the SRA networks and 20,418 genes in the GTEx networks. Our filtering steps are intentionally relaxed, to retain as much data as possible without keeping large amounts of completely uninformative data.

Calculating gene counts

Recount2 stores quantified expression as base pair counts per gene. We converted these values into gene counts by dividing these base pair per gene counts by the average read length in the sample and accounted for paired-end read samples by further dividing by a factor of two.

Refine.bio data collection and processing

To evaluate the workflows on RNA-Seq data processed with different read alignment and counts quantification methods, we matched as many SRA datasets in our final recount2 data corpus as possible to data in refine.bio. In some cases, not every sample in a recount2 dataset was available in the refine.bio database. If the number of missing samples dropped the dataset to less than 5 samples, we did not use that dataset to construct a network. This procedure brought the total number of usable refine.bio datasets to 188, most of which (120/188) contained all of the samples that were used in the recount2 datasets. These datasets were downloaded from refine.bio as unnormalized transcript counts. Because some data in refine.bio was aligned using Ensembl release 93 and the rest was aligned using Ensembl release 96, we first subsetted all refine.bio transcripts to only the common transcripts between releases. The transcript counts were summed to gene counts (using Ensembl release 96 and the biomaRt R package [25], then subset to genes present in the recount2 data. Once gene counts are calculated, the rest of each workflow was run exactly the same as it was on the recount2 datasets.

Within-sample normalization

Within-sample normalization is designed to transform the expression levels of genes within the same sample so that they can be compared to each other. Here, we considered counts per million (CPM), transcripts per million (TPM), and reads per kilobase million (RPKM) for performing within-sample normalization of the original raw gene counts [28, 50]. Note that RPKM is almost the same as fragments per kilobase million (FPKM), except FPKM was introduced to accommodate paired-end RNA-seq so it accounts for the fact that two reads can map back to a single fragment. We account for paired-end samples with FPKM, but use the term "RPKM" throughout the manuscript. These three ways of normalizing counts are very commonly used in RNA-seq analysis and account for library size and gene/transcript length in different ways. CPM corrects for library size (expressed in million counts) so that each count is expressed as a proportion of the total number reads in the sample. TPM and RPKM are similar methods that correct for both library size and gene length. Each gene count is divided by both the length of the gene and the sum of counts in the sample, but these operations are

done in a different order. TPM divides counts by gene length (in kb) first to get transcript counts and then by total number of transcripts in the sample, resulting in each normalized sample having the same number of total counts. This is not guaranteed for RPKM since it corrects each gene count for the total number of reads in the sample before correcting for gene length.

Between-sample normalization

Between-sample normalization transforms the expression levels of genes across a group of samples so that gene counts from the same gene in different samples can be more accurately compared to each other. We tested quantile (QNT), trimmed mean of *M* values (TMM) [51], and upper quartile (UQ) normalizations [13]. In addition, we tested simple counts adjustment methods we call Counts adjusted with TMM Factors (CTF) and Counts adjusted with Upper quartile Factors (CUF). Quantile normalization is an extremely popular between-sample normalization for microarray samples. Applied to RNA-seq data, QNT forces the distribution of all gene expression values to be exactly the same in each sample. We performed quantile normalization on counts, CPM, TPM, and RPKM using the *preprocessCore* package available from Bioconductor, which implements the quantile normalization described in Bolstad et al. [52]. TMM normalizes across samples by finding a subset of genes whose variation is mostly due to technical rather than biological factors, i.e., not differentially expressed, then using this subset to calculate a scaling factor to adjust each sample. In brief, each sample is compared to a chosen reference sample. A certain upper and lower percentage of data based on absolute intensity and log-fold-change relative to the reference sample is removed (by default, 5% for absolute intensity and 30% for log-fold-change) and the log-fold-changes of the remaining set of genes are used to calculate a single scaling factor for the non-reference samples. UQ normalization first removes all zero-count genes and calculates a scaling factor for each sample to match the 75% quantile of the counts in all the samples. In both TMM and UQ, the scaling factors are made to multiply to one before they are used to adjust the library sizes of each sample. These adjusted library sizes are then used in place of the original library size for a calculation otherwise identical to CPM. We used the *edgeR* package [53] to calculate TMM and UQ scaling factors. These factors were also used for CTF and CUF, respectively, where they served as a divisor for each gene count in the proper sample.

Gene type filtering

We chose to keep only long RNA gene types (mRNA (protein-coding), lncRNA, antisense RNA) as those are the most common gene types used in coexpression analysis and shorter reads make mapping and identification more difficult [54, 55]. The excluded gene types (mostly short RNAs) are also unlikely to show up in our functional gold standard as there is very little functional information about these gene types. Therefore, relationships between genes of these types are harder to evaluate.

Data transformation

A log transformation is standard procedure when working with RNA-seq data, as the expected variance grows with the mean for gene counts [35]. A pseudocount is added to the gene count before taking the log. We use the hyperbolic arcsine (asinh) transformation, which is defined along the entire real number line and circumvents the need for predefining a constant pseudocount and instead calculates a "pseudocount" that is proportional to the original gene count. The asinh function compresses smaller values of x less than a function like the natural logarithm [36, 37].

We also compared asinh to variance stabilizing transformation (VST) [38] and regularized logarithm transformation (rlog) [35] implemented in the DESeq2 R package. These were tested on the GTEx and SRA datasets, except for the six largest GTEx datasets due to the prohibitively long running time of the rlog transformation.

Network construction

A coexpression network was constructed for each individual dataset by calculating the Pearson correlation coefficient between every pair of genes in that dati 'aset using the *Distancer* tool in the *Sleipnir* C++ library [56]. These correlations were treated as the edge weight between gene pairs. We chose Pearson correlation as it has been repeatedly shown to provide a robust measure of gene-gene correlations, especially in small-to-medium-sized datasets that are produced by individual laboratories [7, 48]. Since Spearman correlation is also popular in coexpression analysis, we compared these two correlation metrics on our top ten workflows and found that Pearson correlation results in more accurate coexpression networks than Spearman correlation in both GTEx and SRA datasets, particularly in ensuring the accuracy of the top-scoring edges (Additional file 1: Fig. S16).

Network transformation

We tested two common methods of network transformation, weighted topological overlap (WTO) [9] and context likelihood of relatedness (CLR) [10], that use different aspects of network topology to correct the raw coexpression network. The general idea of WTO is to increase the edge weight between gene pairs that share a high number of network neighbors while diminishing edge weight between gene pairs that are tightly connected to very different sets of genes in the network. All edges in the resulting network will have normalized weighted between zero and one. CLR reweights the edge for each gene pair (*i*, *j*) based on how different the original weight of that edge is relative to all of the connections to gene *i* and all connections to gene *j* (to the rest of the genes in the network). For instance, CLR will upweight an edge between two genes if the edge weight is high compared to all of the other connections of both genes. WTO was implemented using the *wTO* function with the "sign" method in the *wTO* package [49], and CLR was implemented using the *Dat2Dab* function in the *Sleipnir* C++ library.

Network evaluation

The goal of coexpression networks is to capture true functional relationships between genes in the cellular context of the original dataset. Therefore, we evaluated the accuracy of each coexpression network by comparing it to two gold standards, one

representing known generic (tissue-naive) functional relationships and the other representing known tissue-aware gene functional relationships. We assembled these gold standards by beginning with a set of manually-selected Gene Ontology Biological Process (GOBP) terms [48, 57] that were deemed to be specific enough to be confident that any genes co-annotated to them could be considered functionally related via experimental follow-up (see *Supplemental Note*). Specifically, curators were considering the question "if unknown gene/protein G were predicted to be annotated to GOBP term T, would that be enough to consider experimentally testing this relationship between G and T?" Then, any pair of genes that were co-annotated to the same specific GOBP term was set as a positive edge in the gold standard. We only used annotations based on experimental (GO evidence codes: EXP, IDA, IPI, IMP, IGI, TAS) or curated evidence (IC). We explicitly ignored gene-term annotations made based on expression (GO evidence code: IEP) to avoid circularity when comparing coexpression-derived interactions to this gold-standard. We next had to determine which pairs of genes among the ones with at least one positive edge could be declared as negative edges, i.e., gene pairs that are unlikely to be functionally related based on prior knowledge. To be clear, "positive" and "negative" are used here based on machine learning parlance to indicate interactions and non-interactions, respectively, and do not correspond to the sign of the relationship. This way, the terms are consistent with how we refer to true/false positive/negative edges. Following previous work, we ignored gene pairs not co-annotated to any specific term but still interact with many of the same genes in the gold standard (determined based on each being annotated to two different terms that contained very similar sets of genes; hypergeometric test; p value <0.05). We also ignored gene pairs that were not co-annotated to any specific term but were co-annotated to certain general GOBP terms, thus introducing ambiguity in whether they are functionally related or not. All remaining gene pairs were considered negatives. We built the naive gold standard using the *Answerer* function in the *Sleipnir* C++ library.

We created the tissue-aware gold standards for as many tissues as possible by subsetting the naive gold standard based on genes known to be specifically expressed in a particular tissue. We obtained tissue-aware genes from the TISSUES 2.0 database Knowledge channel [58]. The knowledge channel contains curated manual annotations of tissue expression provided by UniProtKB. For a given tissue, a positive edge from the naive gold standard was kept in its tissue-aware standard if both genes were expressed in that tissue. Negative edges were kept if both genes were expressed in that tissue, or if one gene is expressed in the tissue and the other gene is expressed in one of the other tissues considered. Only standards containing at least 50 positive edges were used for evaluation, resulting in 24 tissue-aware gold-standards. We specifically excluded epithelium from consideration for a tissue-aware standard, as there is no straightforward way to determine the body site each sample was taken from.

We used the *DChecker* function in the *Sleipnir* C++ library to compare each coexpression network to each gold-standard and return the number of true positives, false positives, true negatives, and false negatives at various edge weight thresholds. These numbers were used to calculate the area under the precision-recall curve (auPRC) using the *trapz* function in the *pracma* package. Since gene functional relationship gold-standards of different tissues have different proportions of positives to negatives, the original auPRC scores are not directly comparable to each other. Therefore, we divided

each auPRC by its "prior"—the auPRC of a random predictor, equal to the fraction of positives among all positive and negative edges—and expressed the performance as the logarithm of this ratio to enable tissue-to-tissue comparisons.

The *Supplemental Note* contains more details on the (i) gene functional relationship gold standard, (ii) additional gold standards that we explored (including spike-ins [59, 60]) and their limitations, and (iii) calculation of the evaluation metrics.

Workflow comparison and analysis by parts

To assess whether two workflows resulted in coexpression networks that were significantly different in quality, we used a paired Wilcoxon rank sum test to compare the auPRC scores across all coexpression networks generated by those two workflows. After calculating p values, we performed a correction for multiple testing with the Benjamini-Hochberg procedure and declared workflows with FDR ≤ 0.01 as being significantly different. Further, each workflow is a combination of method choices at multiple stages. So, to determine the impact of including a particular method in a workflow, we across aggregated workflows to calculate the proportion of times that including a particular method in a workflow resulted in the workflow being significantly greater than one that did not include the method. As it is not possible to do within-sample normalization and then do TMM, UQ, CTF, or CUF, any workflow including CPM, TPM, or RPKM was excluded when assessing between-sample normalization methods so that method being compared to each other based on the same number of aggregated workflows. For similar reasons, workflows involving TMM, UQ, CTF, or CUF were not considered for the analysis of within-sample normalization methods.

GTEx resampling

To simulate uniformly processed datasets that have sample sizes similar to datasets from SRA, we chose nine sample sizes (5, 6, 7, 9, 11, 13, 16, 25, and 40) based on the distribution of SRA dataset sample sizes. Then, from each GTEx dataset with at least 70 samples, we randomly sampled a "dataset" of each sample size, repeating this sampling ten times to create 10 datasets per sample size from each GTEx dataset. One coexpression network was constructed and evaluated from each of these GTEX-resampled datasets in the same manner outlined above.

Experimental factor analysis

In addition to dataset size (i.e., number of samples), the quality of the coexpression network reconstructed from a dataset could also depend on the similarity between the samples in that dataset as well as the total number of mapped reads. We performed an analysis to determine this impact using the GTEx-resampled datasets and the original SRA datasets. Since SRA datasets are not large enough to do resampling for sample size analysis, we split them into five groups with equal number of datasets, with datasets in each group having similar sample sizes. We define sample similarity for a given dataset as the median spearman correlation between all samples using the 50% most variable genes in the GTEx tissue they came from for the resampled GTEx datasets, or the median spearman correlation between all samples using the 50% most variable genes in each individual dataset in the case of the SRA networks. Read count diversity is

calculated by summing the gene counts of each sample in a given dataset and taking their standard deviation. Based on each of these measures—sample similarity and read count diversity—we divided the datasets into five groups of equal size while taking care to check that each group contained datasets with similar sample sizes. For the tissue analysis, we could only determine significance in SRA tissues that had at least 15 datasets.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-021-02568-9.

Additional file 1: Supplemental Figures and a Note. This file contains **supplemental figures S1-S16** and a Supplemental Note describing various aspects of the gold standard.

Additional file 2. Review history.

Acknowledgements

We thank the members of the Krishnan Lab for helpful discussion. We are particularly grateful to Anna Yannakopoulos and Chris Mancuso for code advice and Stephanie Hickey for suggestions on the manuscript.

Peer review information

Barbara Cheifet was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history

The review history is available as Additional file 2.

Authors' contributions

KAJ and AK designed the study. KAJ performed all the analyses. KAJ and AK interpreted the results and wrote the final manuscript. The authors read and approved the final manuscript.

Author's information

Twitter handles: @kaylainbio (Kayla A Johnson); @compbiologist (Arjun Krishnan)

Funding

This work was primarily supported by the US National Institutes of Health (NIH) grants R35 GM128765 to A.K. and supported in part by MSU start-up funds to A.K.

Availability of data and materials

The expression datasets used in this study can be obtained from the recount2 database https://jhubiostatistics.shinyapps.io/recount/. All the results from this study are available at https://krishnanlab.github.io/RNAseq_coexpression [61]. All coexpression networks in this study constructed using the top-performing workflows are available at https://doi.org/10.5281/zenodo.5510567 [62].

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 28 September 2020 Accepted: 6 December 2021
Published online: 03 January 2022

References

1. van Dam S, Võsa U, van der Graaf A, Franke L, Magalhães D, Pedro J. Gene co-expression analysis for functional classification and gene–disease predictions. Brief Bioinform. 2018;19:575–92.
2. Allocco DJ, Kohane IS, Butte AJ. Quantifying the relationship between co-expression, co-regulation and gene function. BMC Bioinformatics. 2004;5:18.
3. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci. Natl Acad Sci. 1998;95:14863–8.
4. Segal E, Friedman N, Koller D, Regev A. A module map showing conditional activity of expression modules in cancer. Nat Genet. 2004;36:1090–8.
5. Carpenter AE, Sabatini DM. Systematic genome-wide screens of gene function. Nat Rev Genet. 2004;5:11–22.

6. Zhang B, Horvath S. A general framework for weighted gene co-expression network analysis. Stat Appl Genet Mol Biol De Gruyter. 2005;4. https://doi.org/10.2202/1544-6115.1128.
7. Zhu Q, Wong AK, Krishnan A, Aure MR, Tadych A, Zhang R, et al. Targeted exploration and analysis of large cross-platform human transcriptomic compendia. Nat Methods. 2015;12:211–4.
8. Evans C, Hardin J, Stoebel DM. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. Brief Bioinform. 2017;19:776–92.
9. Nowick K, Gernat T, Almaas E, Stubbs L. Differences in human and chimpanzee gene expression patterns define an evolving network of transcription factors in brain. Proc Natl Acad Sci. 2009;106:22358–63.
10. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, et al. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol. 2007;5. https://doi.org/10.1371/journal.pbio.0050008.
11. Reverter A, Barris W, McWilliam S, Byrne KA, Wang YH, Tan SH, et al. Validation of alternative methods of data normalization in gene co-expression studies. Bioinforma Oxford Acad. 2005;21:1112–20.
12. Lim WK, Wang K, Lefebvre C, Califano A. Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. Bioinformatics Oxford Acad. 2007;23:i282–8.
13. Bullard JH, Purdom E, Hansen KD, Dudoit S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. BMC Bioinformatics. 2010;11:94.
14. Maza E, Frasse P, Senin P, Bouzayen M, Zouine M. Comparison of normalization methods for differential gene expression analysis in RNA-Seq experiments. Commun Integr Biol. 2013;6. https://doi.org/10.4161/cib.25849.
15. Dillies M-A, Rau A, Aubert J, Hennequet-Antier C, Jeanmougin M, Servant N, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Brief Bioinform Oxford Acad. 2013;14:671–83.
16. Zyprych-Walczak J, Szabelska A, Handschuh L, Górczak K, Klamecka K, Figlerowicz M, et al. The impact of normalization methods on RNA-Seq data analysis. BioMed Res Int. 2015. https://doi.org/10.1155/2015/621690.
17. Abbas-Aghababazadeh F, Li Q, Fridley BL. Comparison of normalization approaches for gene expression studies completed with high-throughput sequencing. PLOS ONE. Public Library of. Science. 2018;13:e0206312.
18. Ballouz S, Verleyen W, Gillis J. Guidance for RNA-seq co-expression network construction and analysis: safety in numbers. Bioinformatics. 2015;31:2123–30.
19. Collado-Torres L, Nellore A, Kammers K, Ellis SE, Taub MA, Hansen KD, et al. Reproducible RNA-seq analysis using recount2. Nat Biotechnol. 2017;35:319–21.
20. Lonsdale J, Thomas J, Salvatore M, Phillips R, Lo E, Shad S, et al. The Genotype-Tissue Expression (GTEx) project. Nat Genet Nat Publ Group. 2013;45:580–5.
21. Leinonen R, Sugawara H, Shumway M. The sequence read archive. Nucleic Acids Res. 2011;39:D19–21.
22. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene Ontology: tool for the unification of biology. Nat Genet. 2000;25:25–9.
23. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE. 2015;10. https://doi.org/10.1371/journal.pone.0118432.
24. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proc 23rd Int Conf Mach Learn. New York: Association for Computing Machinery; 2006. p. 233–40.
25. Greene CS, Hu D, Jones RWW, Liu S, Mejia DS, Patro R, et al. refine.bio: a resource of uniformly processed publicly available gene expression datasets. https://www.refine.bio.
26. Oshlack A, Wakefield MJ. Transcript length bias in RNA-seq data confounds systems biology. Biol Direct. 2009;4:14.
27. Huang J, Vendramin S, Shi L, McGinnis KM. Construction and optimization of a large gene coexpression network in maize using RNA-Seq data. Plant Physiol Am Soc Plant Biologists. 2017;175:568–83.
28. Wagner GP, Kin K, Lynch VJ. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. Theor Biosci. 2012;131:281–5.
29. Zhao S, Ye Z, Stanton R. Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols. RNA. 2020. https://doi.org/10.1261/rna.074922.120.
30. Hicks SC, Irizarry RA. quantro: a data-driven approach to guide the choice of an appropriate normalization method. Genome Biol. 2015;16:117.
31. Cosgrove EJ, Gardner TS, Kolaczyk ED. On the choice and number of microarrays for transcriptional regulatory network inference. BMC Bioinformatics. 2010;11:454.
32. Wang Y, Hicks SC, Hansen KD. Co-expression analysis is biased by a mean-correlation relationship. bioRxiv 2020.02.13.944777; https://doi.org/10.1101/2020.02.13.944777.
33. Farahbod M, Pavlidis P. Differential coexpression in human tissues and the confounding effect of mean expression levels. Bioinforma Oxford Acad. 2019;35:55–61.
34. Ramsköld D, Wang ET, Burge CB, Sandberg R. An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. PLOS Comput Biol. 2009;5:e1000598.
35. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014;15:550.
36. Johnson NL. Systems of frequency curves generated by methods of translation. Biometrika [Oxford Univ Press, Biometrika Trust]. 1949;36:149–76.
37. Hoffman MM, Buske OJ, Wang J, Weng Z, Bilmes JA, Noble WS. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. Nat Methods Nat Publ Group. 2012;9:473–6.
38. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol. 2010;11:R106.
39. Risso D, Ngai J, Speed TP, Dudoit S. Normalization of RNA-seq data using factor analysis of control genes or samples. Nat Biotechnol Nat Publ Group. 2014;32:896–902.
40. Parsana P, Ruberman C, Jaffe AE, Schatz MC, Battle A, Leek JT. Addressing confounding artifacts in reconstruction of gene co-expression networks. Genome Biol. 2019;20:94.
41. Jaffe AE, Hyde T, Kleinman J, Weinbergern DR, Chenoweth JG, McKay RD, et al. Practical impacts of genomic data "cleaning" on biological discovery using surrogate variable analysis. BMC Bioinformatics. 2015;16:372.

42. Mao W, Rahimikollu J, Hausler R, Chikina M. DataRemix: a universal data transformation for optimal inference from gene expression datasets. Bioinformatics. 2021;37(7):984–91.
43. Zhang Y, Cuerdo J, Halushka MK, McCall MN. The effect of tissue composition on gene co-expression. Brief Bioinform. 2021;22(1):127–39.
44. Farahbod M, Pavlidis P. Untangling the effects of cellular composition on coexpression analysis. Genome Res. 2020;30: 849–59.
45. Crow M, Paul A, Ballouz S, Huang ZJ, Gillis J. Exploiting single-cell expression to characterize co-expression replicability. Genome Biol. 2016;17:101.
46. Harris BD, Crow M, Fischer S, Gillis J. Single-cell co-expression analysis reveals that transcriptional modules are shared across cell types in the brain. Cell Syst. 2021;12(7):748–56.e3.
47. Vieth B, Parekh S, Ziegenhain C, Enard W, Hellmann I. A systematic evaluation of single cell RNA-seq analysis pipelines. Nat Commun. 2019;10:1–11.
48. Greene CS, Krishnan A, Wong AK, Ricciotti E, Zelaya RA, Himmelstein DS, et al. Understanding multicellular function and disease with human tissue-specific networks. Nat Genet. 2015;47:569–76.
49. Gysi DM, Voigt A, Fragoso T, de M, Almaas E, Nowick K. wTO: an R package for computing weighted topological overlap and a consensus network with integrated visualization tool. BMC Bioinformatics. 2018;19:1–6.
50. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat Methods Nat Publ Group. 2008;5:621–8.
51. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol. 2010;11:R25.
52. Bolstad BM, Irizarry RA, Åstrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics. 2003;19:185–93.
53. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26:139–40.
54. Garber M, Grabherr MG, Guttman M, Trapnell C. Computational methods for transcriptome annotation and quantification using RNA-seq. Nat Methods Nat Publ Group. 2011;8:469–77.
55. Łabaj PP, Leparc GG, Linggi BE, Markillie LM, Wiley HS, Kreil DP. Characterization and improvement of RNA-Seq precision in quantitative transcript expression profiling. Bioinformatics. 2011;27:i383–91.
56. Huttenhower C, Schroeder M, Chikina MD, Troyanskaya OG. The Sleipnir library for computational functional genomics. Bioinformatics. 2008;24:1559–61.
57. Myers CL, Barrett DR, Hibbs MA, Huttenhower C, Troyanskaya OG. Finding function: evaluation methods for functional genomic data. BMC Genomics. 2006;7:187.
58. Palasca O, Santos A, Stolte C, Gorodkin J, Jensen LJ. TISSUES 2.0: an integrative web resource on mammalian tissue expression. Database J Biol Databases Curation. 2018;2018:bay003. https://doi.org/10.1093/database/bay003.
59. McCall MN, Almudevar A. Affymetrix GeneChip microarray preprocessing for multivariate analyses. Brief Bioinform. 2012; 13:536–46.
60. Qing T, Yu Y, Du T, Shi L. mRNA enrichment protocols determine the quantification characteristics of external RNA spike-in controls in RNA-Seq studies. Sci China Life Sci. 2013;56:134–42.
61. Johnson KA, Krishnan A. RNAseq_coexpression. Github. https://github.com/krishnanlab/RNAseq_coexpression. 2021.
62. Johnson KA, Krishnan A. Coexpression networks of 31 GTEx and 256 SRA RNA-Seq datasets. Zenodo. https://zenodo.org/record/5510567#.YZ1IrfHMJTY. 2021.

Publisher's Note