



Article

GCEN: An Easy-to-Use Toolkit for Gene Co-Expression Network Analysis and lncRNAs Annotation

Wen Chen ^{1,2} , Jing Li ², Shulan Huang ¹, Xiaodeng Li ¹, Xuan Zhang ² , Xiang Hu ¹, Shuanglin Xiang ^{1,*} and Changning Liu ^{2,3,4,*}

¹ State Key Laboratory of Developmental Biology of Freshwater Fish, School of Life Sciences, Hunan Normal University, Changsha 410081, China; chenwen@biochen.org (W.C.); huangshulan5920@163.com (S.H.); deng18397646718@163.com (X.L.); huxiang@hunnu.edu.cn (X.H.)

² CAS Key Laboratory of Tropical Plant Resources and Sustainable Use, Xishuangbanna Tropical Botanical Garden, Chinese Academy of Sciences, Kunming 650223, China; lijing3@xtbg.ac.cn (J.L.); zhangxuan@xtbg.ac.cn (X.Z.)

³ Core Botanical Gardens, Center of Economic Botany, Chinese Academy of Sciences, Menglung 666303, China

⁴ The Innovative Academy of Seed Design, Chinese Academy of Sciences, Kunming 650223, China

* Correspondence: xsplin@hunnu.edu.cn (S.X.); liuchangning@xtbg.ac.cn (C.L.)

Abstract: Gene co-expression network analysis has been widely used in gene function annotation, especially for long noncoding RNAs (lncRNAs). However, there is a lack of effective cross-platform analysis tools. For biologists to easily build a gene co-expression network and to predict gene function, we developed GCEN, a cross-platform command-line toolkit developed with C++. It is an efficient and easy-to-use solution that will allow everyone to perform gene co-expression network analysis without the requirement of sophisticated programming skills, especially in cases of RNA-Seq research and lncRNAs function annotation. Because of its modular design, GCEN can be easily integrated into other pipelines.



Citation: Chen, W.; Li, J.; Huang, S.; Li, X.; Zhang, X.; Hu, X.; Xiang, S.; Liu, C. GCEN: An Easy-to-Use Toolkit for Gene Co-Expression Network Analysis and lncRNAs Annotation. *Curr. Issues Mol. Biol.* **2022**, *44*, 1479–1487. <https://doi.org/10.3390/cimb44040100>

Academic Editor: Md Mehedi Hasan

Received: 6 February 2022

Accepted: 23 March 2022

Published: 25 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: gene co-expression network; lncRNAs annotation; RNA-Seq; gene ontology; KEGG

1. Introduction

Long non-coding RNAs (lncRNAs) are generally defined as transcripts with more than 200 nucleotides but without protein-coding potential. A variety of lncRNAs play import roles in diverse biological processes, such as human diseases [1], innate immunity [2], abnormal development [3], and so on. The recent advances in RNA-Seq technology have immensely boosted the discovery of abundant lncRNAs across many species. The characterization of the functions of lncRNAs are accruing but still lag behind expectations. Unlike protein-coding genes, lncRNAs are often less conserved at the level of primary sequence [4]. As a result, traditional gene annotation approaches, which are based on the similarity of sequence or structure and have achieved great success in protein-coding genes, have become impracticable in the exploration of lncRNAs function.

In contrast, the guilt-by-association approaches, which use associations or interactions between genes to extract functional meaning rather than sequence similarity analysis, provide a feasible strategy for lncRNAs function annotation [5]. The gene co-expression network is one such method which is generally constructed according to high-throughput gene expression profiles from microarray or RNA-Seq. The assumption of gene annotation by a co-expression network is that genes with similar expression pattern are likely to share similar functions [6]. Existing studies have revealed that lncRNAs often execute functions with their protein partners, and coding-lncRNA gene co-expression network analysis has been widely used to identify the potential functions of novel lncRNAs in human [7] and in many model organisms [5,8].

At present, there are some tools or web servers that provide gene co-expression network analysis and/or lncRNAs function annotation services. For example, WGCNA is

an R-based tool for gene co-expression network construction and module identification mainly for microarray data [9]. The server ncFANs is a co-expression-network based webservice for lncRNAs function annotation [10]. However, there are certain new situations to consider for RNA-Seq and coding-lncRNA gene co-expression network analysis, such as data normalization and network construction methods [11], as well as the computational challenges of much larger datasets when lncRNA genes are considered. In addition, it is valuable to run such calculations efficiently on a local computer in many scenarios. Thus, we require an effective cross-platform toolkit for providing all services with network construction and module identification, which has been proven to be effective in many applications on function prediction and annotation of lncRNAs [5,8].

In this work, we present GCEN, a cross-platform command-line toolkit developed with C++, which can be used by biologists to easily build a gene co-expression network and predict gene function. The toolkit GCEN is an efficient and easy-to-use solution that can be easily integrated into other pipelines due to its modular design. It will allow everyone to perform gene co-expression network analysis without the requirement of sophisticated programming skills, especially in RNA-Seq research or in lncRNAs function annotation.

2. Materials and Methods

The program GCEN was developed using C++ as an open-source software under the GPLv3 license. In addition to our code, we used some third-party code in compliance with their licenses. The program GCEN can be compiled and run in Linux, Windows, and macOS. Compiling GCEN requires a compiler and library support for the ISO C++ 2011 standard. No new computational methods or algorithms are used in GCEN, rather we implemented and integrated some classic algorithms to annotate lncRNA faster and more easily. We describe the main algorithms used as follows.

2.1. Data Normalization

We implemented five algorithms, namely, median normalization, quantile normalization [12], the median-of-ratios method [13], trimmed mean of M-values (TMM) [14], and housekeeping genes normalization. For the median normalization, the median of gene expression values in each sample is calculated, and then each gene expression value in the same sample is multiplied by the mean of all medians and divided by the median of this sample. For housekeeping genes normalization, we adjusted the gene expression values so that the median of the housekeeping genes expression in each sample was the same. The descriptions of the other three algorithms can be found in their original papers.

2.2. Co-Expression Network Construction

We used the Spearman or Pearson correlation coefficients directly to determine co-expression patterns between gene pairs, which are better performing and more robust [11]. The coefficient of Spearman or Pearson correlation ρ was calculated as follows.

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad (1)$$

For Pearson correlation, x or y represents the vector of the expression value of each gene, x_i or y_i stands for each expression value, and \bar{x} or \bar{y} , is the mean value of these expression values. For Spearman correlation, the vector of the expression value of each gene needs to be replaced by their ranks. So, Spearman correlation is more robust than Pearson correlation in the face of outliers and is recommended to be used in RNA-Seq data [15].

2.3. Module Identification

The network modules, which are groups of genes with similar expression profiles, are explored based on the topological structure of the gene co-expression network [16]. Genes

in the same module tend to be functionally related and co-regulated. We implemented a module identification algorithm that was based on the node similarity measure of their relative interconnectedness coupled with the hierarchical clustering method [17].

$$s_{ij} = \frac{h_{ij} + a_{ij}}{c_i + c_j - h_{ij} - a_{ij}} \quad (2)$$

where s_{ij} is the similarity between gene i and gene j , h_{ij} is the number of shared neighbour genes of gene i and gene j in the co-expression network, a_{ij} is the adjacency of gene i and gene j , and c_i and c_j are the connectivity of gene i and gene j , respectively.

2.4. Function Annotation

After gene co-expression network construction and module identification, we used gene function enrichment to predict the function of novel lncRNAs or coding genes. The p value of the enriched function was calculated as follows.

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i} \binom{N-M}{n-i}}{\binom{N}{n}} \quad (3)$$

where N is the total number of background genes, M is the number of genes with one certain annotation in background genes, n is the number of neighbours of the gene to be annotated, and k is the number of neighbour genes with the certain annotation.

Another gene function analysis algorithm we implemented is the random walk with restart (RWR) [18], which measures each node's relevance with respect to given seed nodes (here are genes with known function annotations) based on network propagation. The information (known gene function) is flowed in the network from seed nodes to nearby nodes until convergence. Finally, ranked information associating genes with a function of interest is attached on the nodes of the network.

$$p_k = \alpha p_0 + (1 - \alpha) W p_{k-1} \quad (4)$$

where p_0 represents our initial information on genes (seed), and W is a gene co-expression matrix, its columns sum was set to 1. The parameter α is the probability that the information restart to the starting node. Repeated iteration of this equation converges to a steady state. p_k is the ranked information that associates genes with a function of interest. The implementation of RWR uses Eigen, which is a C++ template library for linear algebra (<http://eigen.tuxfamily.org>, accessed on 12 December 2022).

3. Results

3.1. The Main Analysis Process of GCEN

The recommended pipeline of GCEN consists of four parts: data pretreatment, network construction, module identification, and function annotation (Figure 1). A README file and sample data are included in the software package, which may be of help to users. Because of its modular design, GCEN can be easily integrated into another pipeline. Moreover, the multithreaded implementation of GCEN makes it fast and efficient for RNA-Seq data.

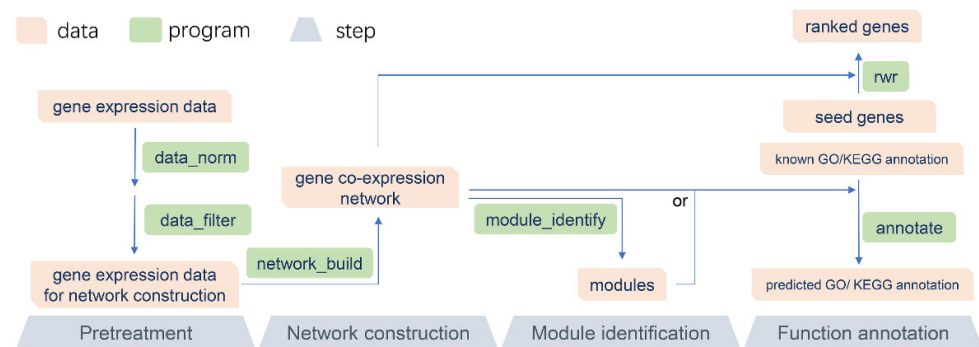


Figure 1. The recommended pipeline of GCEN. The recommended pipeline consists of four parts: data pretreatment, network construction, module identification, and function annotation.

Performing a gene co-expression network analysis requires gene expression profiles, which are usually derived from microarray or RNA-Seq. Before gene co-expression network construction, it is critical that the expression values are normalized on the same measurement scale. To remove systematic effects in the RNA-Seq data, we implemented many state-of-the-art data normalization algorithms, including quantile normalization [12], the median-of-ratios method [13], trimmed mean of M-values (TMM) [14], and housekeeping genes normalization, which have already been used in the gene differential expression analysis. Moreover, low expression of genes will bring noise to the gene co-expression network, and therefore, we needed to remove these genes according to the mean or variance of their expression values.

Next, according to the guidelines of RNA-Seq co-expression network construction and analysis, we used the Spearman or Pearson correlation coefficients directly to determine co-expression patterns between gene pairs, which perform better and are more robust [11]. Then, network modules, which are groups of genes with similar expression profiles, were explored based on the topological structure of the gene co-expression network [16]. Genes in the same module tend to be functionally related and co-regulated. We implemented a module identification algorithm based on the node similarity measure of their relative interconnectedness coupled with the hierarchical clustering method [17].

After gene co-expression network construction and module identification, we used gene function enrichment [19] to predict novel lncRNAs or coding gene function. According to the guilt by association (GBA) principle, we determined the function of an unknown gene by voting on its neighbour genes with known functions. These neighbour genes can be directly interacting genes in the network (Figure 2a), or they can be genes in the same module (Figure 2b). Another gene function analysis algorithm we implemented is the random walk with restart (RWR) (Figure 2c) [18], which measures each node's relevance with respect to given seed nodes (here are genes with known function annotations) based on network propagation. The information (known gene function) is flowed in the network from seed nodes to nearby nodes until convergence. Finally, ranked information that associates genes with a function of interest is attached on the nodes of the network.

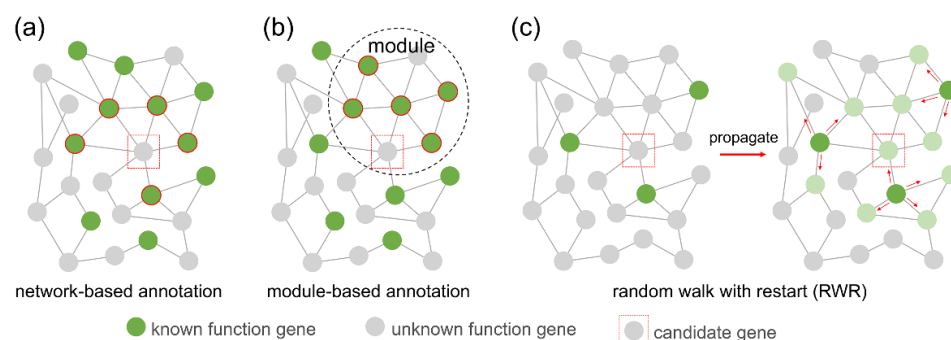


Figure 2. Gene function annotation methods. (a) Network-based function annotation. The function of an unknown gene (red box) is inferred from the function of its neighbours (green background red circle) in the network. (b) Module-based function annotation. The function of an unknown gene (red box) is inferred from the function of other genes (green background red circle) in the module. (c) Random walk with restart (RWR). Information flows in the network from the seed node (dark green) until convergence. Each node has traces of information. The rank of the node (light green) highly associated with the seed node is higher.

3.2. Performance Evaluation

Considering the recent discovery of tens of thousands of lncRNAs transcribed from the genomes of mammals and other complex organisms, we find that the number of nodes and connections in the coding-lncRNA gene co-expression network will be greatly increased in comparison with a traditional coding gene co-expression network. This poses a big challenge to the computation time and memory for constructing a co-expression network. The program GCEN is developed with C++ and has a natural advantage in terms of its performance. We tested the time and memory consumption of GCEN, FastGCN [20], and WGCNA [9] in network construction. The fastest was GCEN and it had the least memory consumption. For a network of 10,000 genes, it only takes a few seconds for network construction, and time consumption may vary because of data size and computer performance. In single thread mode, GCEN is approximately five times faster than WGCNA in calculating the correlation coefficient of gene expression. The outputs of GCEN are generated immediately after the calculation of network construction, and therefore, its memory consumption remains low without increasing significantly with the number of genes. However, WGCNA had a peak memory of more than 20GB when 40,000 genes were analysed (Table 1).

Table 1. Time and memory consumption tests for network construction.

Gene Number	GCEN	FastGCN	WGCNA
10k	9.51 s/5.93 MiB	16.98 s/1.31 GiB	59.36 s/1.84 GiB
20k	37.86 s/8.50 MiB	2 m 11.59 s/5.25 GiB	3 m 47.15 s/6.36 GiB
40k	2 m 31.42 s/12.88 MiB	24 m 23.33 s/21.12 GiB	14 m 57.86 s/24.39 GiB
80k	10 m 7.70 s/21.58 MiB	Out of maximum memory	59 m 53.82 s/96.11 GiB

These tests were run on a personal computer with an Intel Core i5-10400 processor (6 cores/12 threads) and 128GB memory. The version of GCEN was 0.5.1, the version of FastGCN is v1.1, and the version of WGCNA was 1.69. The test data were a randomly generated number between 0 and 1, and each gene has 20 expression values. All tests were run in single thread mode, although GCEN has implemented multi-threading. The test data and scripts can be found on our website (<https://www.biochen.com/gcen/static/benchmark.zip>, accessed on 13 March 2022).

3.3. Data Visualization

Data visualization is essential in biological research because it presents the meaning of the data more straightforwardly. The release of GCEN does not include a plotting program as GCEN focuses on efficiently using a gene co-expression network to predict gene function. However, we did not ignore the plotting demands of biologists, and we provide a number of data visualization demos and scripts on our website. We do not offer automatic plotting instead of enlightenment to show biological significance in gene co-expression analysis.

Figure 3 shows three examples of visualizations. Figure 3a shows the degree distribution of a gene co-expression network. The degree distribution of gene co-expression networks is similar to that of scale-free networks and approximately follows a power law. The majority of genes are related to a small number of other genes, while only a few genes are linked to a large number of genes. This degree distribution can be shown as a straight line on a log-log plot. Figure 3b shows the sub-network or module. We predict gene function on the basis of neighbouring genes in the network or genes in the same module. As a result, it is critical to demonstrate them graphically. Figure 3c shows GO annotations of a gene. Among the three aspects of Gene Ontology, the biological process is the most representative of gene functions and is also the most demonstrated. The code and sample data for Figure 3 are available on our website (<https://www.biochen.org/gcen/visualization>, accessed on 13 March 2022).

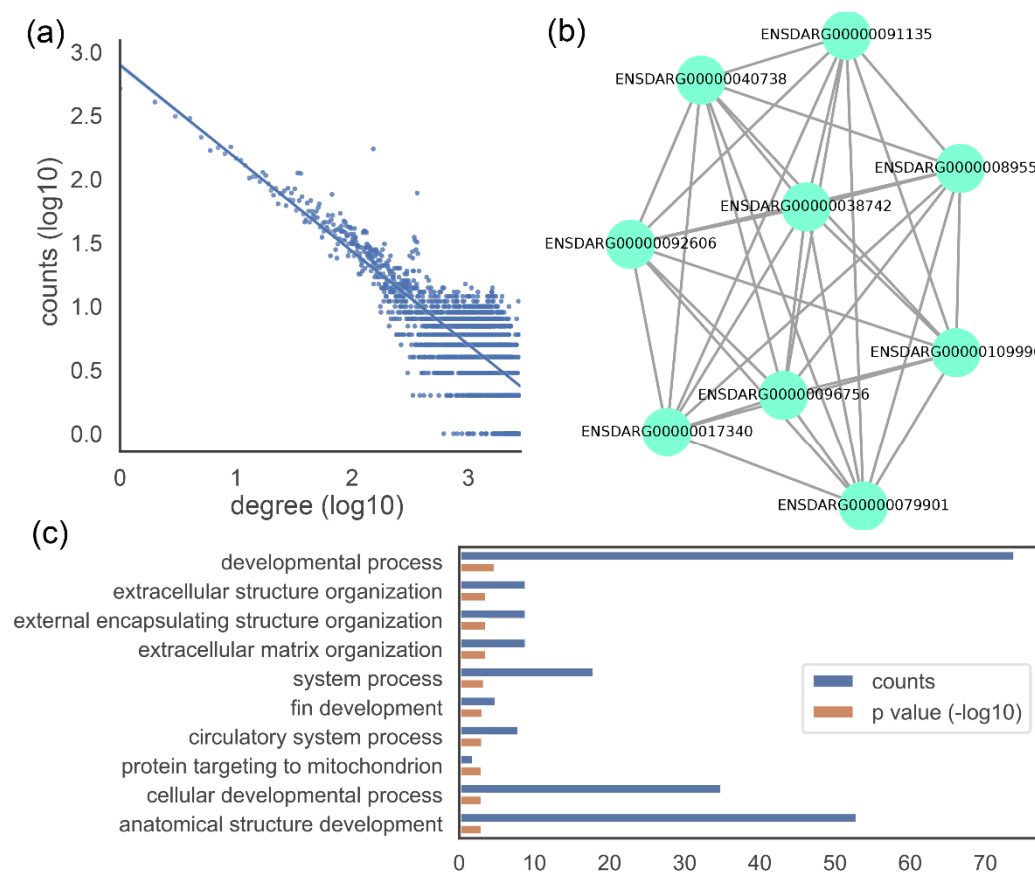


Figure 3. Examples of data visualization. (a) Network degree distribution. (b) Sub-network or module. (c) GO annotation (biological process top 10). The dataset derived from an RNA-Seq of 17 samples of zebrafish [21].

4. Discussion

RNA-Seq is a popular approach for studying gene expression under various biological conditions. Normalization, which scales the raw data so that different samples can be compared, is a critical step in RNA-Seq. Different normalization methods are based on different assumptions that significantly affect the downstream analysis [22]. The toolkit GCEN implements five widely used normalization methods for users to choose from. To solve the performance problem caused by the larger network with many genes in RNA-Seq, we developed GCEN in C++ for performance. A gene co-expression network analysis using GCEN can be carried out on a typical desktop or laptop computer.

The construction of a gene co-expression network is usually achieved by calculating Pearson or Spearman correlation coefficients, such as FastGCN [20], SWIM [23], and SEa-

CorAI [24]. WGCNA [9] first calculated the Pearson correlation coefficient, and then set a soft threshold by selecting the β parameter. Gene pairs with correlation coefficients reaching a threshold were considered to be co-expressed. Similar to WGCNA, CEMiTool [25] modified the method of selecting the β parameter, and GWENA [26] used the Spearman correlation coefficients instead of Pearson correlation coefficients. Ballouz et al. evaluated the performance of network building methods and concluded that the computationally simplest and most transparent method (direct use of the Spearman or Pearson correlation coefficients) for calculating co-expression was better than the soft thresholding method (e.g., used by WGCNA) [11]. Therefore, at present, GCEN can only build the network by calculating the Spearman or Pearson correlation coefficients, and other network building methods will be added in the future.

False positives in gene co-expression networks are a problem that cannot be ignored. Low expression of genes will bring noise to the gene co-expression network, and an empirical approach is to discard a quarter of the total number of genes. Petti et al. found that negative correlations between gene pairs may not be biologically meaningful, and further they developed the SEaCorAI to reduce spurious correlations [24]. It can be considered to keep only positive correlation in gene co-expression network. Ballouz et al. and Franziska et al. figured out that network aggregation shows better performance [11,27]. Thus, a program `network_merge` is included in the GCEN package that can merge two or more networks.

Our research interest focused on using computational approaches to decipher the function of long non-coding RNAs. The toolkit GCEN was also developed with this purpose in mind. Using the gene co-expression network, we obtained 7345 lncRNAs with GO annotation and 7055 lncRNAs with KEGG annotation among 13,604 zebrafish lncRNAs in our previous study [8]. More recently, we annotated 196 of 756 conserved lncRNAs in 25 flowering plants using the same method [28]. Some biologists have had concerns about the annotation accuracy. We are convinced that the prediction accuracy of the real network was higher than the random shuffled network that was used as a control [8]. There is a program `calculate_accuracy` in the GCEN package to calculate the accuracy of prediction. We had developed some utilities like the `calculate_accuracy` to assist in the gene co-expression network analysis.

Our previous study used the gene co-expression network to annotate lncRNAs on a large scale, but the gene co-expression network has a much broader application. Mathias et al. built lncRNAs co-expression networks using TCGA breast cancer data. Furthermore, they found an oncogenic lncRNA LINC00504 [29]. In another study that searched for disease-related genes, Bayraktar identified key genes of Alzheimer's disease (AD) through AD-specific co-expression networks and genome-scale metabolic modelling of the brain in AD patients [30]. The toolkit GCEN is also suitable for such similar analysis. However, there is still some analysis that GCEN does not support, such as identification of hub genes. Recently, Paola Paci et al. developed a new algorithm called SWIM (SWItch Miner) for the discovery of a new class of hub genes from gene co-expression networks. These hub gene are called switch genes and are negatively correlated with their first nearest neighbours [24]. Then, they applied SWIM methodology to identifies novel candidate disease genes [31]. The software SWIM was originally written in MATLAB, but they have recently re-implemented it using R [32].

The toolkit GCEN still has some limitations. The command line does scare off some users. In the future, we will develop a web server based on GCEN to provide a graphical interface. At the same time, we will enhance data visualization in gene co-expression analysis. Currently, GCEN only implements two methods of network construction and one method of module identification. We will add more algorithms in the future.

Software should be well maintained during its lifecycle, and its robustness and ease of use should be improved. The software GCEN will be continuously updated on our website (<https://www.biochen.org/gcen>, accessed on 13 March 2022), and there is also a large amount of material on the website that can assist biologists in using it better. Moreover,

connecting with our users is truly important to us. We sincerely hope that users will provide their questions and suggestions to help us improve the software.

5. Conclusions

We present here GCEN, an easy-to-use toolkit for gene co-expression network analysis and lncRNAs annotation. It has three notable features: it is easy to use, it has a high speed and low memory usage, and it has cross-platform application. The toolkit GCEN was primarily designed to be used in lncRNAs annotation, but is not limited to these scenarios. In the future, we will update the software and fix bugs in accordance with user feedback. We promise to maintain GCEN for 5 years or more. The software GCEN is available on our website (<https://www.biochen.com/gcen>, accessed on 5 February 2022) or on GitHub (<https://github.com/wen-chen/gcen>, accessed on 5 February 2022).

Author Contributions: Conceptualization, W.C., S.X. and C.L.; methodology, W.C. and C.L.; software, W.C.; validation, X.L., X.Z. and X.H.; formal analysis, W.C.; investigation, W.C., S.H. and J.L.; resources, S.X. and C.L.; data curation, X.L., X.Z. and X.H.; writing—original draft preparation, W.C.; writing—review and editing, J.L., S.X. and C.L.; visualization, W.C.; supervision, S.X. and C.L.; project administration, W.C.; funding acquisition, S.X. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (31770109, 81972642, and 31970609), the Opening Fund of State Key Laboratory of Developmental Biology of Freshwater Fish (2020KF003), the Start-up Fund from Xishuangbanna Tropical Botanical Garden, and the “Top Talents Program in Science and Technology” from Yunnan Province. The APC was funded by the Opening Fund of State Key Laboratory of Developmental Biology of Freshwater Fish (2020KF003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: GCEN is freely available under the GPLv3 license at <https://www.biochen.com/gcen> or <https://github.com/wen-chen/gcen>.

Acknowledgments: We appreciate the users who have chosen and trusted GCEN, and who have provided feedback to help us improve it. We also appreciate all colleagues in our laboratory for helpful discussions and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, J.; Xuan, Z.; Liu, C. Long non-coding RNAs and complex human diseases. *Int. J. Mol. Sci.* **2013**, *14*, 18790–18808. [[CrossRef](#)] [[PubMed](#)]
2. Xie, Y.; Wei, Y. A Novel Regulatory Player in the Innate Immune System: Long Non-Coding RNAs. *Int. J. Mol. Sci.* **2021**, *22*, 9535. [[CrossRef](#)] [[PubMed](#)]
3. Schonrock, N.; Harvey, R.P.; Mattick, J.S. Long noncoding RNAs in cardiac development and pathophysiology. *Circ. Res.* **2012**, *111*, 1349–1362. [[CrossRef](#)]
4. Ulitsky, I. Evolution to the rescue: Using comparative genomics to understand long non-coding RNAs. *Nat. Rev. Genet.* **2016**, *17*, 601–614. [[CrossRef](#)]
5. Liao, Q.; Liu, C.; Yuan, X.; Kang, S.; Miao, R.; Xiao, H.; Zhao, G.; Luo, H.; Bu, D.; Zhao, H.; et al. Large-scale prediction of long non-coding RNA functions in a coding-non-coding gene co-expression network. *Nucleic Acids Res.* **2011**, *39*, 3864–3878. [[CrossRef](#)]
6. Van Dam, S.; Vosa, U.; van der Graaf, A.; Franke, L.; de Magalhaes, J.P. Gene co-expression analysis for functional classification and gene-disease predictions. *Brief Bioinform.* **2018**, *19*, 575–592. [[CrossRef](#)] [[PubMed](#)]
7. Cogill, S.B.; Wang, L. Co-expression Network Analysis of Human lncRNAs and Cancer Genes. *Cancer Inf.* **2014**, *13*, 49–59. [[CrossRef](#)]
8. Chen, W.; Zhang, X.; Li, J.; Huang, S.; Xiang, S.; Hu, X.; Liu, C. Comprehensive analysis of coding-lncRNA gene co-expression network uncovers conserved functional lncRNAs in zebrafish. *BMC Genom.* **2018**, *19*, 112. [[CrossRef](#)]
9. Langfelder, P.; Horvath, S. WGCNA: An R package for weighted correlation network analysis. *BMC Bioinform.* **2008**, *9*, 559. [[CrossRef](#)]

10. Liao, Q.; Xiao, H.; Bu, D.; Xie, C.; Miao, R.; Luo, H.; Zhao, G.; Yu, K.; Zhao, H.; Skogerbo, G.; et al. ncFANs: A web server for functional annotation of long non-coding RNAs. *Nucleic Acids Res.* **2011**, *39*, W118–W124. [[CrossRef](#)]
11. Ballouz, S.; Verleyen, W.; Gillis, J. Guidance for RNA-seq co-expression network construction and analysis: Safety in numbers. *Bioinformatics* **2015**, *31*, 2123–2130. [[CrossRef](#)] [[PubMed](#)]
12. Ritchie, M.E.; Phipson, B.; Wu, D.; Hu, Y.; Law, C.W.; Shi, W.; Smyth, G.K. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **2015**, *43*, e47. [[CrossRef](#)] [[PubMed](#)]
13. Love, M.I.; Huber, W.; Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **2014**, *15*, 550. [[CrossRef](#)] [[PubMed](#)]
14. Robinson, M.D.; Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* **2010**, *11*, R25. [[CrossRef](#)] [[PubMed](#)]
15. Hou, J.; Ye, X.; Feng, W.; Zhang, Q.; Han, Y.; Liu, Y.; Li, Y.; Wei, Y. Distance correlation application to gene co-expression network analysis. *BMC Bioinform.* **2022**, *23*, 81. [[CrossRef](#)] [[PubMed](#)]
16. Saelens, W.; Cannoodt, R.; Saeys, Y. A comprehensive evaluation of module detection methods for gene expression data. *Nat. Commun.* **2018**, *9*, 1090. [[CrossRef](#)]
17. Wang, Z.; San Lucas, F.A.; Qiu, P.; Liu, Y. Improving the sensitivity of sample clustering by leveraging gene co-expression networks in variable selection. *BMC Bioinform.* **2014**, *15*, 153. [[CrossRef](#)]
18. Cowen, L.; Ideker, T.; Raphael, B.J.; Sharan, R. Network propagation: A universal amplifier of genetic associations. *Nat. Rev. Genet.* **2017**, *18*, 551–562. [[CrossRef](#)]
19. Klopfenstein, D.V.; Zhang, L.; Pedersen, B.S.; Ramirez, F.; Warwick Vesztrocy, A.; Naldi, A.; Mungall, C.J.; Yunes, J.M.; Botvinnik, O.; Weigel, M.; et al. GOATOOLS: A Python library for Gene Ontology analyses. *Sci. Rep.* **2018**, *8*, 10872. [[CrossRef](#)]
20. Liang, M.; Zhang, F.; Jin, G.; Zhu, J. FastGCN: A GPU accelerated tool for fast gene co-expression networks. *PLoS ONE* **2015**, *10*, e0116776. [[CrossRef](#)]
21. Pauli, A.; Valen, E.; Lin, M.F.; Garber, M.; Vastenhouw, N.L.; Levin, J.Z.; Fan, L.; Sandelin, A.; Rinn, J.L.; Regev, A.; et al. Systematic identification of long noncoding RNAs expressed during zebrafish embryogenesis. *Genome Res.* **2012**, *22*, 577–591. [[CrossRef](#)] [[PubMed](#)]
22. Evans, C.; Hardin, J.; Stoebel, D.M. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. *Brief Bioinform.* **2018**, *19*, 776–792. [[CrossRef](#)] [[PubMed](#)]
23. Liesecke, F.; De Craene, J.O.; Besseau, S.; Courdavault, V.; Clastre, M.; Vergès, V.; Papon, N.; Giglioli-Guivarc’h, N.; Glévarec, G.; Pichon, O.; et al. Improved gene co-expression network quality through expression dataset down-sampling and network aggregation. *Sci. Rep.* **2019**, *9*, 14431. [[CrossRef](#)]
24. Paci, P.; Colombo, T.; Fiscon, G.; Gurtner, A.; Pavesi, G.; Farina, L. SWIM: A computational tool to unveiling crucial nodes in complex biological networks. *Sci. Rep.* **2017**, *7*, 44797. [[CrossRef](#)] [[PubMed](#)]
25. Petti, M.; Verrienti, A.; Paci, P.; Farina, L. SEaCorAl: Identifying and contrasting the regulation-correlation bias in RNA-Seq paired expression data of patient groups. *Comput. Biol. Med.* **2021**, *135*, 104567. [[CrossRef](#)]
26. Russo, P.S.T.; Ferreira, G.R.; Cardozo, L.E.; Burger, M.C.; Arias-Carrasco, R.; Maruyama, S.R.; Hirata, T.D.C.; Lima, D.S.; Passos, F.M.; Fukutani, K.F.; et al. CEMiTool: A Bioconductor package for performing comprehensive modular co-expression analyses. *BMC Bioinform.* **2018**, *19*, 56. [[CrossRef](#)]
27. Lemoine, G.G.; Scott-Boyer, M.P.; Ambroise, B.; Perin, O.; Droit, A. GWENA: Gene co-expression networks analysis and extended modules characterization in a single Bioconductor package. *BMC Bioinform.* **2021**, *22*, 267. [[CrossRef](#)]
28. Sang, S.; Chen, W.; Zhang, D.; Zhang, X.; Yang, W.; Liu, C. Data integration and evolutionary analysis of long non-coding RNAs in 25 flowering plants. *BMC Genom.* **2021**, *22*, 739. [[CrossRef](#)]
29. Mathias, C.; Groeneveld, C.S.; Trefflich, S.; Zambalde, E.P.; Lima, R.S.; Urban, C.A.; Prado, K.B.; Ribeiro, E.; Castro, M.A.A.; Gradia, D.F.; et al. Novel lncRNAs Co-Expression Networks Identifies LINC00504 with Oncogenic Role in Luminal A Breast Cancer Cells. *Int. J. Mol. Sci.* **2021**, *22*, 2420. [[CrossRef](#)]
30. Bayraktar, A.; Lam, S.; Altay, O.; Li, X.; Yuan, M.; Zhang, C.; Arif, M.; Turkez, H.; Uhlén, M.; Shoaie, S.; et al. Revealing the Molecular Mechanisms of Alzheimer’s Disease Based on Network Analysis. *Int. J. Mol. Sci.* **2021**, *22*, 11556. [[CrossRef](#)]
31. Paci, P.; Fiscon, G.; Conte, F.; Wang, R.S.; Farina, L.; Loscalzo, J. Gene co-expression in the interactome: Moving from correlation toward causation via an integrated approach to disease module discovery. *NPJ Syst. Biol. Appl.* **2021**, *7*, 3. [[CrossRef](#)] [[PubMed](#)]
32. Fiscon, G.; Paci, P. SWIMmeR: An R-based software to unveiling crucial nodes in complex biological networks. *Bioinformatics* **2021**, *38*, 586–588. [[CrossRef](#)]