

RNASeqR: An R Package for Automated Two-Group RNA-Seq Analysis Workflow

Kuan-Hao Chao¹, Yi-Wen Hsiao¹, Yi-Fang Lee¹, Chien-Yueh Lee¹, Liang-Chuan Lai¹,
Mong-Hsun Tsai¹, Tzu-Pin Lu¹, and Eric Y. Chuang¹

Abstract—RNA-Seq analysis has revolutionized researchers’ understanding of the transcriptome in biological research. Assessing the differences in transcriptomic profiles between tissue samples or patient groups enables researchers to explore the underlying biological impact of transcription. RNA-Seq analysis requires multiple processing steps and huge computational capabilities. There are many well-developed R packages for individual steps; however, there are few R/Bioconductor packages that integrate existing software tools into a comprehensive RNA-Seq analysis and provide fundamental end-to-end results in pure R environment so that researchers can quickly and easily get fundamental information in big sequencing data. To address this need, we have developed the open source R/Bioconductor package, RNASeqR. It allows users to run an automated RNA-Seq analysis with only six steps, producing essential tabular and graphical results for further biological interpretation. The features of RNASeqR include: six-step analysis, comprehensive visualization, background execution version, and the integration of both R and command-line software. RNASeqR provides fast, light-weight, and easy-to-run RNA-Seq analysis pipeline in pure R environment. It allows users to efficiently utilize popular software tools, including both R/Bioconductor and command-line tools, without predefining the resources or environments. RNASeqR is freely available for Linux and macOS operating systems from Bioconductor (<https://bioconductor.org/packages/release/bioc/html/RNASeqR.html>).

Index Terms—RNA-Seq, analysis workflow, pipeline, R, bioconductor, transcriptome assembly, differential expression analysis, gene expression, statistical analysis, visualization

1 INTRODUCTION

RNA-SEQ is a revolutionary approach with which to discover and investigate an entire transcriptome using next-generation sequencing (NGS) technologies [1]. The rapid advances in massive parallel RNA-Seq, as well as the decrease in cost and the development of various convenient analysis tools [2], have made RNA-Seq a widely used method for producing comprehensive transcriptional information. Typically, the main objective of this kind of

transcriptome analysis is to identify genes that are differentially expressed under different conditions or in different tissues in order to gain an understanding of the physiological pathways associated with pathological conditions [1].

Many transcriptome analyses are case-control studies, i.e., observational studies focusing on an “exposure” (i.e., a risk factor or a medical treatment) which is correlated with a certain outcome [3]. A case-control experiment involves a group of samples with the outcome and a corresponding group without the outcome; it is retrospective in that it traces the different outcomes back to different exposures [3]. Examples of transcriptional case-control studies include the analysis of lesional (case) and normal (control) samples extracted via skin biopsies from *Homo sapiens* for increasing the understanding of psoriasis [4], and a comparison of the differential lipid metabolism gene expression between unparasitized *Aphids gossypii* and *Aphids gossypii* parasitized by *Lysiphlebia japonica* [5]. From these studies, we see that comparative RNA-Seq analysis is widely applicable and may play an important role in clinical or biological studies.

There are many well-developed algorithms and statistical methods for each analysis step of RNA-Seq, including aligners, transcript assemblers, and statistical methods for analyzing the differential expression of genes or transcripts in terms of either read counts or FPKM-based expression values [6]. However, this kind of piecemeal RNA-Seq analysis involves many software tools written in different programming languages for each processing step. It will be more efficient to integrate the individual analysis steps together into an automated workflow. The R programming language and its associated Bioconductor project have become popular

- K. Chao is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan. E-mail: b05901180@ntu.edu.tw.
- Y. Hsiao and C. Lee are with the Bioinformatics and Biostatistics Core, Center of Genomic and Precision Medicine, National Taiwan University, Taipei 10617, Taiwan. E-mail: y.w.hsiao9419@gmail.com, d00945006@ntu.edu.tw.
- Y. Lee is with the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei 10617, Taiwan. E-mail: r05945010@ntu.edu.tw.
- L. Lai is with the Graduate Institute of Physiology, National Taiwan University, Taipei 10617, Taiwan. E-mail: llai@ntu.edu.tw.
- M. Tsai is with the Institute of Biotechnology, National Taiwan University, Taipei 10617, Taiwan. E-mail: motion@ntu.edu.tw.
- T. Lu is with the Institute of Epidemiology and Preventive Medicine, Department of Public Health, National Taiwan University, Taipei 10617, Taiwan. E-mail: tphu@ntu.edu.tw.
- E. Y. Chuang is with the Bioinformatics and Biostatistics Core, Center of Genomic and Precision Medicine, National Taiwan University, Taipei 10617, Taiwan and Biomedical Technology and Device Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan, and also with the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei 10617, Taiwan. E-mail: chuangey@ntu.edu.tw.

Manuscript received 31 May 2019; revised 26 Oct. 2019; accepted 22 Nov. 2019. Date of publication 2 Dec. 2019; date of current version 7 Oct. 2021.

(Corresponding author: Eric Y. Chuang.)

Digital Object Identifier no. 10.1109/TCBB.2019.2956708

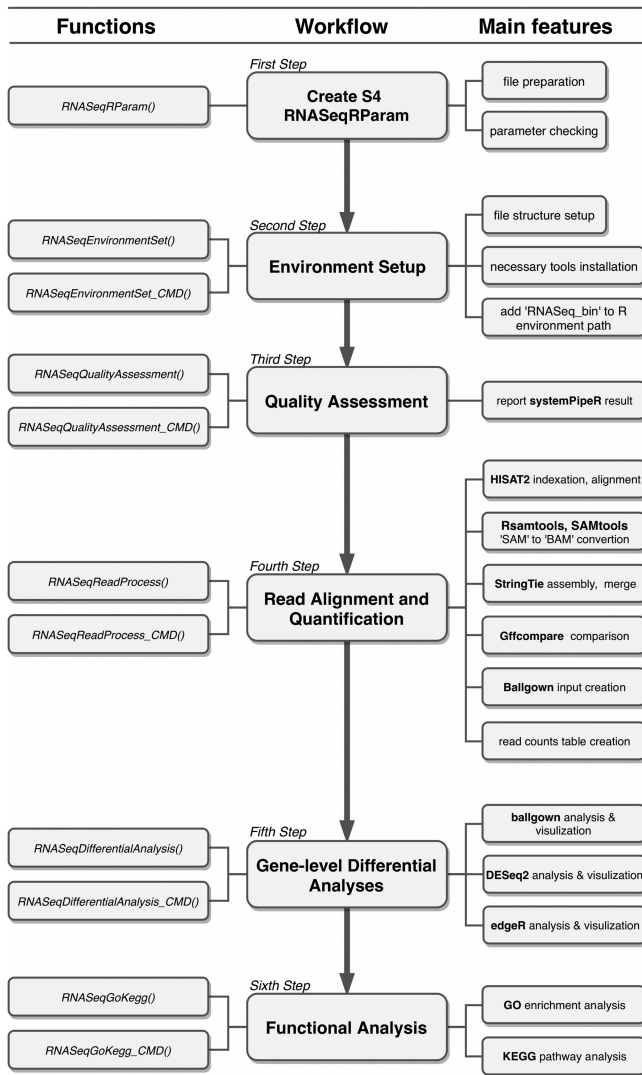


Fig. 1. Overview of the full workflow of RNASeqR. There are three parts in this figure: the main workflow (middle), the functions involved in each step (on the left), and the substeps that can be performed within each function (on the right). Aside from the function *S4 RNASeqRParam*, the remaining functions can be conducted in either the R-interactive version or the background version (by adding the *CMD* suffix). After correctly running the functions in order, a two-group comparative RNA-Seq data analysis will be done.

and are widely used in many fields of biological research. While Bioconductor provides substantial R packages for certain parts of the overall pipeline of RNA-Seq analysis [7], there are few R packages for performing the analysis process in a comprehensive and reproducible way. Thus, an RNA-Seq analysis package within R will be beneficial for experimental or clinical exports.

RNASeqR was designed to offer an automated RNA-Seq workflow for running a transcriptome analysis pipeline in six steps with the support of comprehensive visualization, the integration of command-line tools in the R environment, and the option for functions operating either in an R-interactive version or in a background version.

2 METHODS

RNASeqR is an open-source Bioconductor package for analyzing two-group RNA-Seq data with at least three replicates in each group. It is designed to simplify the analysis

procedure by implementing a standardized workflow [8], which provides reproducible and trustworthy results in the R environment. RNASeqR is available on Linux and macOS operating systems, with support for integrating the command-line interface tools HISAT2 [9], STAR [10], StringTie [11], and Gffcompare into the R environment, and is able to be run in an interactive R shell (R-interactive version) or in the background (background version). By running well-constructed functions in RNASeqR, the results files created from each step, in either tabular or graphical format, are kept in a well-defined file structure.

2.1 Workflow Design

The RNASeqR workflow (Fig. 1) is designed to conduct a two-group RNA-Seq data analysis in only six steps, the main design concept being that each step is implemented as an R function. This six-step procedure includes: (1) the creation of an *S4 RNASeqRParam* object, (2) the setup of the environment, (3) read quality assessment among all samples, (4) read alignment and quantification, (5) gene-level differential analyses, and (6) functional analysis. Users have to run these functions in order, as some steps will process only when the input file generated from the previous step exists. Before running subsequent steps, users must make sure the current step has finished successfully, whether in the R-interactive version or the background version. Some steps take about 6 hours to finish; therefore, it is highly recommended to run the background version. Moreover, an *All_Steps_Interface* function which wraps steps from Quality assessment to functional analysis is also defined. Thus, users are able to execute the workflow in a single function call which makes the pipeline more automatic. If the program terminates accidentally, users do not need to go into the project folder and remove any files. Simply rerun the function and everything will be redone. The only exception is the third step, Quality Assessment, which is an optional step that users can run any time they want.

3 RESULTS

To illustrate the utility of RNASeqR, the package workflow was applied to a set of raw RNA-Seq data. Six samples of *Saccharomyces cerevisiae* RNA were selected from NCBI's Sequence Read Archive (SRA) [12]: entries SRR3396381, SRR3396382, SRR3396384, SRR3396385, SRR3396386, and SRR3396387 in BioProject PRJNA318684. Suitable reference genome and gene annotation files in Ensembl version R64-1-1 were downloaded from the iGenomes Illumina support site (https://support.illumina.com/sequencing/sequencing_software/igenome.html). Three samples (SRR3396381, SRR3396382, SRR3396384) were treated with 0.03 $\mu\text{g}/\text{mL}$ amphotericin B and are referred to as the case group, while the remaining samples (SRR3396385, SRR3396386, SRR3396387) did not receive any treatment and were the control group. Both sets of yeast cultures were grown in ID20 medium (inhibition of cell growth by 20 percent) for 60 minutes [13].

3.1 RNASeqRParam Object Creation

A *S4* class named *RNASeqRParam* was designed to contain all of the essential data required for transcriptome analysis. Users have to prepare an *input_files* directory containing raw files formatted in accordance with the file structure regulation

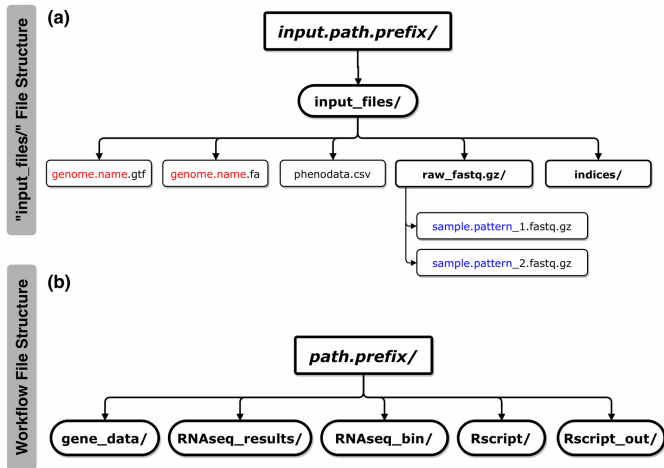


Fig. 2. a The structure of the *input_files* directory. *input.path.prefix* is the parameter of the absolute path in which the *input_files* directory is located. Under the *input_files* directory, there are several necessary files that users have to provide in order to pass the checking procedure of the *RNASeqRParam* constructor function: (i) *genome.name.gtf*, (ii) *genome.name.fa*, (iii) *phenodata.csv*, (iv) the *raw_fastq.gz* directory containing raw paired-end (2 files) or single-end (1 file) fastq.gz files of all samples and (v) the *indices* directory (optional) for storing HISAT2 or STAR indexes. The *genome.name* and *sample.pattern* files marked in red and blue, respectively, are the constructor function parameter values. b The structure of the *RNASeqR* root directory. *path.prefix* is the absolute path of the *RNASeqR* root directory. Five main directories are created in the second step, Environment Setup. (i) *gene_data* contains symbolic links to gtf, fa, fastq.gz, and ht2 files and intermediate files in bam, sam, etc.; (ii) *RNAseq_results* stores important *RNASeqR* outputs including alignment, differential gene expression, functional results in tabular csv format, visualization plots as well as a COM-MAND.txt with executed commands; (iii) *RNAseq_bin* contains HISAT2, STAR, StringTie, and Gffcompare binaries in compressed and decompressed forms; (iv) *Rscript* contains R scripts that are created by running background version functions (with the CMD suffix); and (v) *Rscript_out* contains Rout log files created by running background version functions.

(Fig. 2a). The consistency between user input parameters of the constructor function *RNASeqRParam()* and raw files in the *input_files* directory will be thoroughly checked, and any mismatch will cause this step to fail. The successfully created *RNASeqRParam* S4 object is used as the input parameter in the subsequent analysis functions, which can simplify the whole RNA-Seq data analysis process. *RNASeqRParam* starts the pipeline with raw fastq.gz files. We also designed *RNASeqRParam_Sam()* and *RNASeqRParam_Bam()* to let users run RNA-Seq analysis from SAM files and BAM files.

3.2 Environment Setup

By running the function for this step, either *RNASeqEnvironmentSet()* or *RNASeqEnvironmentSet_CMD()*, the basic file structure of *RNASeqR* (Fig. 2b) is created, and the necessary third-party command-line interface software programs used in this workflow, e.g., HISAT2, STAR, StringTie, and Gffcompare, are downloaded automatically. The *RNAseq_bin* directory containing these decompressed binaries is automatically added to the R environment path variable. *RNAseq_results* directory stores all important *RNASeqR* results including statistical analyses, visualization plots and a COMMAND.txt storing executed commands

3.3 Quality Assessment

Quality assessment of raw reads is an important step in RNA-Seq data analysis, leading to high quality outcomes. This

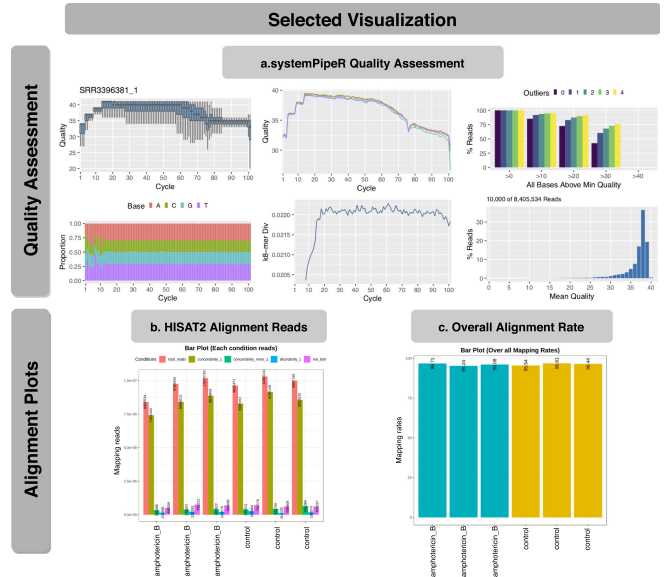


Fig. 3. Selected visual results of the Quality Assessment and Read Alignment steps. a Quality assessment results reported by the systemPipeR Bioconductor package providing sequence quality score distribution, GC content, read length distribution, etc. b HISAT2 alignment results for each sample: (i) total reads, (ii) aligned concordantly exactly 1 time, (iii) aligned concordantly > 1 time, (iv) aligned discordantly exactly 1 time, and (v) reads not aligned. c The overall alignment rate of each sample. Blue and yellow represent two distinct groups.

mainly includes calculating the sequence quality score and GC content, and checking for the presence of the adapter sequence [14]. After running *RNASeqQualityAssessment()* or *RNASeqQualityAssessment_CMD()*, the quality report of all samples is created by systemPipeR [15] and kept in the *RNASeq_results* directory (Fig. 3a). The assigned score of each base call, also called the Phred score, refers to the probability of a base being incorrect [16]. Higher value indicates a higher chance that a call is correct. Generally, it is recommended that the average score of each sample should at least reach Q30, which means each base call has a 1 in 1000 probability of being incorrect (i.e., 99.9 percent base call accuracy) [17]. In RNA-SeqR, quality trimming is not included due to the limitations on trimming large RNA-Seq fastq.gz files in the R environment. We suggest omitting samples with an average quality score below Q30 from the subsequent analyses. To let users update trimmed fastq.gz files easily, an update function, *Update_Fastq_gz*, is provided. In the case of our *S. cerevisiae* RNA-Seq experiment, the average score of all samples was over Q30, indicating the quality of this set of sequencing data was sufficient to proceed with the alignment step (Fig. 3a).

3.4 Read Alignment and Quantification

Reads are typically mapped to an annotated genome, and then we quantify the number of reads that are assigned to a specific gene region based on the comprehensive gene annotation file. A correct reference genome sequence and accurate gene annotation lead to accurate estimation of gene expression. This essential and all-in-one step involves reference indexing (HISAT2 or STAR), read alignment (HISAT2 or STAR), SAM-to-BAM file conversion (Samtools), transcript assembly (StringTie), transcript comparison (Gffcompare), ballgown input creation, and read count table creation (StringTie). Users can alter the parameterization of HISAT2, STAR

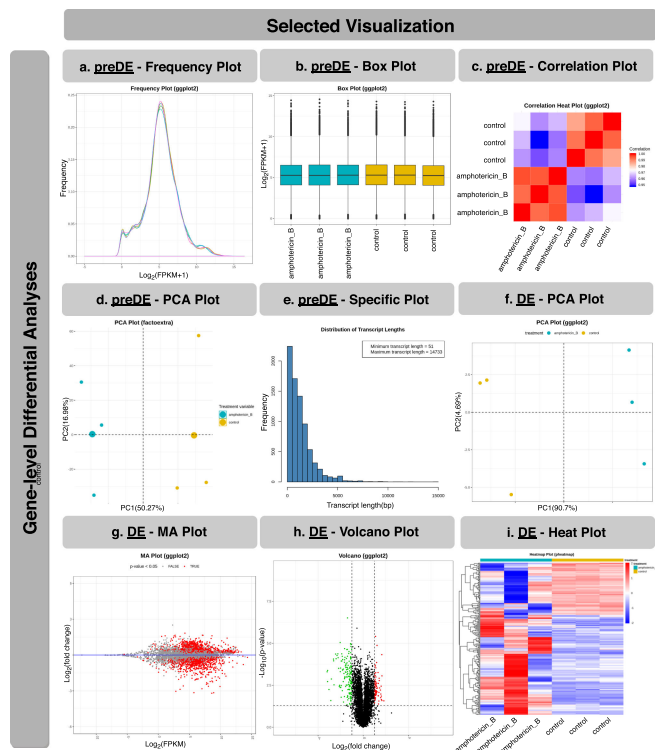


Fig. 4. Selected visual results of gene-level differential analyses. a Frequency plot of normalized gene expression. Each curve represents one sample. b Box plot of normalized gene expression. c Principal Component Analysis (PCA) plot of all samples. The x-axis is the most dominant principal component (PC1); the y-axis is the second most dominant principal component (PC2). d Correlation heatmap plot with the depth of color representative of correlation between samples. Red and blue represent the maximum and minimum correlation values, respectively. e Distribution plot of transcript length from the ballgown package. Package-specific plots are available for each differential analysis tool. f PCA plot of all samples using the expression values of DEGs. g Mean average (MA) plot. Each dot corresponds to a gene. Genes with a p-value smaller than 0.05 are marked in red. h Volcano plot. Each dot represents a DEG. Down-regulated and up-regulated genes, based on default filter criteria, are marked in green and red, respectively. i Sample-feature (Gene) heatmaps using k-means clustering on gene expression. The log₂-normalized (FPKM+1) values of all the genes were used. The two sample classes (i.e., case and control) are indicated in blue and yellow, respectively.

and StringTie to customize their analyses. Default parameters will be applied if users do not specify customized parameters.

All of these substeps are completed after running `RNASeqReadProcess()` or `RNASeqReadProcess_CMD()`. However, this step takes considerable time to finish as it contains several substeps. Taking the *S. cerevisiae* RNA-Seq experiment as an example, six FASTQ files, each with 8.4-10.3 million reads, were processed using 16 CPU cores and 160 GB of RAM; this consumed 14,531.328 seconds for the user CPU time and 4,184.594 seconds for the system CPU time (> 5 hours in total). Therefore, `RNASeqReadProcess_CMD()`, which creates an R script and processes in the background, is highly recommended for this step. Alternatively, under the premise that certain substeps are already successfully finished, users can skip them by setting the corresponding function parameter in order to speed up the whole process.

The percentage of mapped reads is an important parameter for the overall alignment accuracy. For a model organism, it is recommended that 70-90 percent of RNA-Seq data should map to its corresponding genome [18]. To check mapping quality,

in our package, different mapping types and overall alignment rates of each sample are summarized as tabular output and visualized using bar charts (Figs. 3b and 3c). Overall, the mapping rates of all the *S. cerevisiae* samples were over 95 percent, illustrating that both the library of sequencing reads and the reference genome for read alignment are highly accurate.

3.5 Gene-Level Differential Analyses

The most common application of RNA-Seq is to identify gene expression changes between experimental conditions, elucidating possible biological mechanisms behind such experimental designs. In this step, three Bioconductor packages, supporting the detection of differentially expressed genes (DEGs), are applied: ballgown [19], DESeq2 [20], and edgeR [21], [22]. The default thresholds of the p-value and log₂ fold change for DEGs are 0.05 and 1, but they can be manually modified for specific needs. RNASeqR provides comprehensive outputs in both tabular and graphical format by running `RNASeqDifferentialAnalysis()` or `RNASeqDifferentialAnalysis_CMD()`. All visual plots created during this process are stored separately in the respective image directories under `ballgown_analysis`, `DESeq2_analysis`, or `edgeR_analysis`.

In each directory, images are divided into two categories: pre-differential expression (`preDE` directory) and differential expression (`DE` directory). Prior to differential expression analysis, the distribution of gene expression of each sample in terms of normalized counts or FPKM value is displayed as frequency plots and box plots (Figs. 4a and 4b) using ggplot2 [23] in order to check whether the normalization approach is suitable for each dataset. For example, in our *S. cerevisiae* case, both plots showed a consistent distribution profile when applying the proper normalization. The pairwise correlations of expression profiles between all samples is visualized as a heatmap (Fig. 4c) using corrrplot [24] and shows how the samples are related to each other. Samples with higher similarity to one another will be nearer each other. This helps users to further identify outlier samples whose correlation with all other samples is low. Likewise, principal component analysis (PCA), a multivariate analysis technique, is used to represent the whole gene expression profile of each sample as a single value [25]. The two-dimensional PCA plot (Fig. 4d) is generated using FactoMineR/factoextra [26] and ggplot2R packages for exploring within- and between-group clustering. It allows users to identify how sample classes separate and exclude the outlier. In the two-group *S. cerevisiae* RNA-Seq case, both the correlation plot and the PCA plot (Figs. 4c and d) revealed that the clustering of case versus control samples was good, without any outliers that should be dropped out. Moreover, RNASeqR also provides package-specific plots for each DE analysis package, such as the distribution of transcript length produced by ballgown, shown in Fig. 4e. After differential expression analysis using three approaches, four common plots are generated in each DE folder: a PCA plot, a mean average plot, a volcano plot, and a heatmap. The PCA plot, at this stage, illustrates whether expression of selected DEGs successfully separates sample classes as well (Fig. 4f). In our *S. cerevisiae* example, the variance explained by the first principal component increases from 50.27 to 90.5 percent, indicating that the expression of those selected DEGs is sufficient to distinguish the difference between conditions. The mean average plot of the log₂ fold change for all detected genes and the volcano

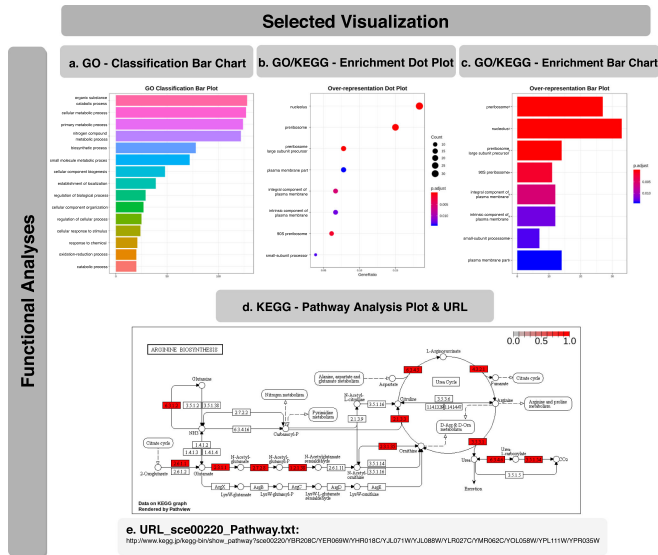


Fig. 5. Selected visual results of functional analyses. a The top 15 GO classification results in terms of molecular function, cellular component, and biological process are visualized using a bar graph. Enrichment results of GO analysis and KEGG pathway analysis are presented using both a bar graph (b) and a dot plot (c). d Visualization of significantly enriched pathway IDs found by KEGG pathway analysis with the pathway Bioconductor package. Pathway ID sce0020 is illustrated as an example and the corresponding URL on the KEGG website is provided in (e).

plot comparing the statistical significance and the fold change for each gene are generated using ggplot2 (Figs. 4g and 4h). Significant DEGs are highlighted in both plots to visualize whether the proportion of DEGs across all detected genes under default cutoffs is appropriate. To further visualize the expression of DEGs in the *S. cerevisiae* samples, a heatmap is also generated using the pheatmap [27] R package (Fig. 4i). It allows the user to identify up-regulated and down-regulated genes in the case group compared with the control.

3.6 Functional Analysis

It is important to understand the biological impact of DEGs involved in the condition of interest. After gene-level differential analyses, users can run `RNASEqGoKegg()` or `RNASEqGoKegg_CMD()` to determine the enriched biological processes/pathways of DEGs found by the analysis using ballgown, DESeq2, and edgeR. The clusterProfiler package [28] used in this step provides two main features: Gene Ontology (GO) [29], [30] analysis and Kyoto Encyclopedia of Genes and Genomes (KEGG) [31], [32] pathway analysis. Both features are integrated into RNASEQR. For GO analysis, classification and over-representation results are presented in terms of three aspects of GO: molecular function, biological process, and cellular component. In the statistical output from DESeq2, for instance, the bar chart for the GO classification of DEGs (Fig. 5a) shows the top 15 biological processes involved in amphotericin B treatment of *S. cerevisiae*. The dot plot (Fig. 5b) and bar chart (Fig. 5c) of GO over-representation illustrate that the nucleolus and preribosome are the two significantly involved cellular components with smallest p-values and highest counts. For KEGG pathway analysis, enriched pathways are stored in csv format and the five most significant pathways are visualized by pathview [33], a Bioconductor package (Fig. 5d). Furthermore, the URL of the pathway is also stored in txt format on the KEGG website

(Fig. 5e). Once this step is successfully finished, a comprehensive RNA-Seq data analysis using RNASEQR is complete.

3.7 More Results From Published Data

Additional two-group RNA-Seq data analysis examples are available. RNA-Seq analysis results for Gene Expression Omnibus (GEO) [34], [35] entries GSE100075 [36] and GSE50760 [37] from *Homo sapiens* using RNASEQR are presented in Additional files 5, 6 and 7.

4 DISCUSSION

There are several reasons that R was chosen as the development platform for RNASEQR. First, R is currently one of the most popular languages for statistical data analysis in bioinformatics and data science. Second, R is an open source and open development language providing simple data manipulation methods and powerful visualization utilities for developers [38]. Third, Bioconductor provides more than 1600 high-quality R software packages for statistical analysis and genomic data processing in each step of a high-throughput method. Moreover, there are also many annotation, experiment, and workflow packages on Bioconductor, which makes the R environment developer-friendly [39], [40]. Fourth, the R language is supported by Bioconda. Bioconda is a bioinformatics channel hosted on GitHub that turns recipes into conda packages, and is maintained by the bioinformatics community [41]. Bioconda provides a nice platform for packages supported by both Linux and macOS operating systems, which makes the installation process much easier.

4.1 Reasons to Implement This Pipeline

RNASEQR provides a fast way for researchers to quickly get overview of sequencing data in pure R environment and its six-step approach is the most convenient and easiest method among all current RNA-Seq pipeline tools; therefore, it would assist clinical researchers without significant computational background to grasp fundamental RNA-Seq results easily.

In RNASEQR, the new tuxedo pipeline published in *Nature Protocols* in 2016 [8] is fully implemented in the R environment, including extra features such as quality assessment, differential expression analysis, and functional analysis. Each tool in RNASEQR is carefully selected. In this pipeline, both command-line interface tools and R packages are used. Conducting an RNA-Seq analysis piecemeal in separate environments using similar repeat commands for each sample is inefficient and monotonous for users. Therefore, RNASEQR is designed to integrate command-line interface tools into an R package and automate each step. The Following are the reasons for selected tools.

HISAT2 is the main alignment tool used in RNASEQR, and it was chosen because of the following advantages. (i) HISAT2 is a highly memory-efficient system using the Burrows-Wheeler transform and the Ferragina-Manzini index. Moreover, it supports genomes of any size, which makes this pipeline widely applicable. (ii) Its processing speed is faster compared with OLego [42], STAR [10], and TopHat2 [43], and therefore it speeds up the whole RNASEQR analysis procedure. (iii) Its sensitivity and precision are higher than other alignment tools. The real data results of 20 million simulated 100-bp reads and small anchor reads revealed that HISAT2 series tools have the greatest accuracy [9]. Spliced Transcripts

Alignment to a Reference (STAR) [10] is another widely used, fast, reference-based aligner that is included in RNASeqR to provide users with another choice in the alignment step.

Regarding transcript assembly and quantification, StringTie and Cufflinks [44] are the two most popular tools in RNA-Seq analysis. StringTie was chosen for use in RNASeqR for the following reasons. (i) StringTie has better performance on identifying the dominant transcript for a gene locus. For example, the number of transcripts only detected by StringTie is higher than those only detected by Cufflinks; therefore, it is better in terms of isoform prediction. (ii) StringTie is superior at reconstructing genes with low abundance, more exons, or multiple isoforms. (iii) StringTie runs faster than Cufflinks and Traph [45]. The lower processing time of StringTie speeds up the RNASeqR pipeline [11].

Ballgown [46] is a statistical package for estimating differentially expressed genes, transcripts, or exons. The benefit of ballgown is that it is highly compatible with the output from StringTie, which is suitable for the RNASeqR workflow. Moreover, two popular Bioconductor tools for count-based differential gene expression analysis, DESeq2 and edgeR, are also implemented in this pipeline, providing results based on different statistical methods.

4.2 R-Interactive Version and Background Version

An important feature in RNASeqR is that it provides two options for running the workflow. Aside from the first step, there are two types of functions for the rest of the steps: R-interactive version and background version. The first one, R-interactive version, can be called and executed in the R shell just like normal functions. However, some functions take a great deal of time and are not practical to run in the R shell. With real 10 versus 10 (20 raw fastq.gz files, roughly 3 GB per file) whole human RNA-Seq data, for example, it took 206,755.42 seconds for the user CPU time and 148,708.59 seconds for the system CPU time under 160 GB of RAM in 16 threads (almost 100 hours total). Therefore, the background version function, which has the `_CMD` suffix, is preferable. By calling this function, a small script will be created and run in the background with `nohup R CMD`. The latest progress of the function will be reported in the Rout file of the `Rscript_out` directory. Users should ensure the function is finished successfully before running subsequent functions.

4.3 Well-Defined File Structure

Because RNASeqR has its own well-established file structure, users only need to run functions in order and do not need to manually organize the file structure by themselves. The root file structure of RNASeqR (Fig. 2b) will be created in the second step, Environment Setup, and all files created in each subsequent step will be stored properly. For example, alignments, such as bam and sam files, and their statistical results will be stored in their corresponding directories; this facilitates the processing of subsequent analyses based on these preliminary files.

4.4 Comprehensive Visualization

Another main feature of RNASeqR is that it provides comprehensive visualized plots for users to further interpret the results. In Gene-level Differential Analysis (the fifth step), the results of ballgown, DESeq2, and edgeR will be created, and

users can manually set the function parameters to skip any differential analysis tool. For the functional analyses, GO and KEGG results are visualized in bar graphs and dot plots. The high-quality figures generated in each step are all in png format at 300 dpi.

4.5 Command-Line Software Integration

RNASeqR integrates many command-line tools, e.g., HISAT2, STAR, StringTie, and Gffcompare, into the R environment. Users do not need to download these tools on their own. Instead, all compressed binaries of these tools will be downloaded automatically during the second step, Environment Setup, based on the operating system that is detected on the workstation. Subsequently, they will be decompressed and stored in the `RNAseq_bin` directory, which will be added to the R environment path. The shell commands of these tools are available by running the `system2()` function in the R environment. Although there are so many substeps in RNA-Seq analysis, the only thing that users need to do is to call the `RNASeqEnvironmentSet()` or `RNASeqEnvironmentSet_CMD()` function and all the relevant tools will be set up accordingly.

4.6 Ease of Use

RNASeqR is an open source tool available on both Bioconductor and Bioconda, providing users with simple installation methods. Moreover, the latest version can also be downloaded directly through GitHub. All these methods allow users to install all dependent R packages with only a single command. Running RNASeqR only requires the fundamentals of the Linux operating system and R language; that is, users only need to run simple R functions in order and check the results files to make sure the process is completely finished for conducting a basic comparative RNA-Seq data analysis.

4.7 Additional Experiment Data Package

Each Bioconductor package needs a comprehensive vignette to demonstrate the usage of each function. RNASeqRData, an additional Bioconductor experiment data package, was created for vignette demonstration of the RNASeqR software package. These RNA-Seq sample data were extracted from the data used in the case study. The original fastq.gz files are about 800 MB, which is too large for a vignette demonstration; therefore, to reduce the size of the files but keep as many differentially expressed genes as possible, only the reads aligned to the region from bases 0 to 100,000 on chromosome XV were extracted.

4.8 Comparison to Other Tools

There are several tools designed for NGS analysis. Galaxy [47], Taverna [48], Snakemake [49], Nextflow [50], CIPHER [51], and bcbio-nextgen [52] all provide NGS data analysis infrastructure in various computer language environments. For example, tools such as VIPER [2], TRAPLINE [53], and HppRNA [54] are RNA-Seq pipelines based on the Snakemake framework, combining tools in numerous languages including R, Python, Perl, C++, or Java. QuickRNASeq [55] is a self-development pipeline run on linux with HTML interactive visualization. Although there are many frameworks for RNA-Seq data analysis, only a few of them are able to run end-to-end RNA-Seq data analysis in a pure R environment. Table 1 provides a comparison of selected RNA-Seq analysis tools.

TABLE 1
RNA-Seq Tools Comparison

	RNA-Seq Software						
	RNASeqR	SystemPipeR	QuasR	bcbioRNASeq	VIPER	QuickRNASeq	HppRNA
Main Feature	Automated six-step RNA-Seq pipeline in R	Automated workflows for NGS in R	Framework for quantification and analysis of Short Reads	R package for bcbio RNA-Seq output data	Visualization snakemake pipeline for RNA-seq	Large scale RNA-seq pipeline with dynamic visualization	Snakemake parameter-free RNA-Seq pipeline
Freeware License	Artistic-2.0	Artistic-2.0	GPL-2	MIT	GNU GPL	GNU GPL	MIT
Installation Repository	Bioconductor, Bioconda	Bioconductor, Bioconda	Bioconductor, Bioconda	GitHub remotes, Bioconda	Bitbucket	sourceforge	sourceforge
Main Environment	R >= 3.5.0	R	R >= 3.6.0	R >= 3.6.0	Conda / Snakemake	Bash, R, Perl, Javascript	Snakemake
System Prerequisite	BiocManager	BiocManager	BiocManager	remotes, bcbio, Python	wget, git, Conda	STAR, Subread, VarScan, RSeQC, Samtools	Python, Perl, R
Automaticity (From documentations)	6 lines of R code from environment setup to functional analysis.	50+ lines of R code for end-to-end RNA-Seq analysis	80+ lines of R code for end-to-end analysis	Separate R functions for each different bcbio outputs.	Several bash commands for self-setup environment.	Several bash commands for self-setup environment.	Several bash commands for self-setup environment.
Quality Assessment	Yes ^a	Yes	Yes	Yes	Yes	Yes	Yes
Alignment / Assembly	Yes ^a	Yes	Yes	No ^e	Yes	Yes	Yes
DE Analysis	Yes ^a	Yes	No	Yes	Yes	Yes	Yes
Visualization	Yes ^b	Yes ^c	Yes ^d	Yes ^f	Yes ^g	Yes ^h	Yes ⁱ
Functional Analysis	Yes ^a	Yes	No	Yes	Yes	No	No

^aOne-line R functions. Both R-interactive version or background version are provided. ^bAlignment report plots, reads frequency plots, box plots, violin plots, pre-differential-expression samples PCA plots, correlation plots, volcano plot, post-differential-expression samples PCA plots, heatmap plots, ballgown, DESeq2, and edgeR specific plots, GO classification bar chart, GO/KEGG enrichment analysis bar charts and pathview pathway plots are provided. ^cCorrelation dendrogram, differential expression genes bar chart, Venn diagram, GO Bar chart, heatmap plot etc. are provided. ^dQuality control report with sequence qualities and alignment statistics plot etc. are provided. ^eAlignment and assembly are included in bcbio python pipeline but not in bcbioRNASeq. ^fQuality control plot, reads alignment plot, 5'->3' bias plot, gene distribution plot, reads box plot, frequency plot, variance stabilization plot, PCA plot and heatmap plot etc. are provided. ^gReads alignment bar chart, read distribution plot, gene body coverage curves & bar chart, heatmap, PCA plot, differential gene expression summary plot, volcano plot, GO & KEGG enrichment analysis plot, box plot, Q-Q plot, fusion-Gene analysis summary plot etc. are provided. ^hQuality control reports with interactive plots, SNP correlation plot, Parallel plots of multi-dimensional QC measures, reads alignment plot, interactive gene expression visualization etc. are provided. ⁱQuality control plot, Heatmap, samples PCA plot etc. are provided.

RNASeqR is implemented as a Bioconductor R package. Bioconductor is a central repository storing bioinformatics-related R packages, which has the following advantages. (i) It resolves dependencies between packages. Installing packages manually might have version incompatibility problems. (ii) It provides reliable code for all packages. Software tools must pass technical review before being accepted, and developers have to provide a comprehensive vignette with runnable example code. (iii) Packages that are out of maintenance are removed regularly.

Most Bioconductor packages are only designed for a specific analysis step or part of the overall pipeline. Take QuasR [56] as an example; it provides a workflow from raw sequence reads to quality assessment and quantification of genomic regions, which are the initial steps of a few NGS pipelines, such as RNA-Seq and Bis-Seq. However, it lacks an interface with external third-party command-line tools and downstream analyses like differential gene expression and pathway analysis. Although some packages provide a comprehensive framework integrating external command-line tools into the R environment for a complete NGS pipeline, these still involve many manual operations for users.

For example, systemPipeR [15] provides a flexible framework for many analysis pipelines, such as RNA-Seq, CHIP-Seq, Ribo-Seq, and VAR-Seq, with external command-line

tools in the R environment. However, the whole analysis pipeline is not completely automated. Users still have to run complex downstream analyses manually using other Bioconductor packages. In systemPipeR RNA-Seq workflow document in Bioconductor, users have to write fifty more lines of code manually in order to do downstream analysis. In contrast, only two functions are needed in RNASeqR in order to finish differential expression, functional analysis and get comprehensive visualization. Furthermore, RNA-SeqR provides both R-version and background-version to process commands.

Similarly, bcbioRNASeq [57], another R package for comprehensive RNA-Seq data analysis, takes bcbio [52] output as input and provides functions for data visualization and downstream analysis. However, users still have to run bcbio and bcbioRNASeq separately and manually with several lines of codes. Unlike these packages, RNASeqR not only provides a comprehensive RNA-Seq data analysis but also focuses on one specific pipeline which has been recently reported as the most efficient and widely-used workflow [8]. Furthermore, RNASeqR largely simplifies RNA-Seq analysis, starting from quality assessment to functional analysis, into just six functions and provides a large number of interpretable plots. Therefore, our package allows clinical researchers to speed up the process of RNA-Seq data

analysis, retaining high accuracy and providing graphical results for further biological interpretation.

5 CONCLUSION

RNASeqR is a new Bioconductor package providing a six-step automated workflow for two-group comparative RNA-Seq analysis. The core design concept of RNASeqR is that each RNA-Seq analysis step is implemented as an R function in the package, and thus users can perform RNA-Seq analysis instinctively. The main features of RNASeqR include: (i) flexible function options in both an R-interactive version and a background version, (ii) comprehensive visualization, (iii) integration of command-line tools into the R environment, (iv) a well-defined file structure, and (v) ease of use. We believe that RNASeqR will assist clinical researchers without significant computational background to obtain useful information from RNA-Seq data in an easy, efficient and reproducible manner.

ACKNOWLEDGMENTS

This work has been supported in part by the Center of Genomic and Precision Medicine, National Taiwan University, Taiwan [106R8400]; and the Center for Biotechnology, National Taiwan University, Taiwan [GTZ300]. The authors thank Melissa Stauffer, Ph. D., for editing the manuscript.

REFERENCES

- Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: A revolutionary tool for transcriptomics," *Nature Rev. Genetics*, vol. 10, no. 1, pp. 57–63, Jan. 2009.
- M. Cornwell *et al.*, "VIPER: Visualization pipeline for RNA-seq, a snakemake workflow for efficient and complete RNA-seq analysis," *BMC Bioinf.*, vol. 19, no. 1, Apr. 12, 2018, Art. no. 135.
- S. Lewallen and P. Courtright, "Epidemiology in practice: Case-control studies," *Community Eye Health*, vol. 11, no. 28, pp. 57–58, 1998.
- B. Li *et al.*, "Transcriptome analysis of psoriasis in a large case-control sample: RNA-seq provides insights into disease mechanisms," *J. Investigative Dermatology*, vol. 134, no. 7, pp. 1828–1838, Jul. 2014.
- G. XueKe, Z. Shuai, L. JunYu, L. LiMin, Z. LiJuan, and C. JinJie, "Lipidomics and RNA-Seq Study of Lipid Regulation in *Aphis gossypii* parasitized by *Lysiphlebia japonica*," *Sci. Rep.*, vol. 7, no. 1, May 2, 2017, Art. no. 1364.
- A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nature Methods*, vol. 5, no. 7, pp. 621–628, Jul. 2008.
- W. Huber *et al.*, "Orchestrating high-throughput genomic analysis with Bioconductor," *Nature Methods*, vol. 12, no. 2, pp. 115–121, Feb. 2015.
- M. Pertea, D. Kim, G. M. Pertea, J. T. Leek, and S. L. Salzberg, "Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown," *Nature Protocol*, vol. 11, no. 9, pp. 1650–1667, Sep. 2016.
- D. Kim, B. Langmead, and S. L. Salzberg, "HISAT: A fast spliced aligner with low memory requirements," *Nature Methods*, vol. 12, no. 4, pp. 357–360, Apr. 2015.
- A. Dobin *et al.*, "STAR: Ultrafast universal RNA-seq aligner," *Bioinf.*, vol. 29, no. 1, pp. 15–21, Jan. 1, 2013.
- M. Pertea, G. M. Pertea, C. M. Antonescu, T. C. Chang, J. T. Mendell, and S. L. Salzberg, "StringTie enables improved reconstruction of a transcriptome from RNA-seq reads," *Nature Biotechnology*, vol. 33, no. 3, pp. 290–295, Mar. 2015.
- Y. Kodama, M. Shumway, R. Leinonen, and C. International Nucleotide Sequence Database, "The sequence read archive: Explosive growth of sequencing data," *Nucl. Acids Res.*, vol. 40, no. Database issue, pp. D54–D56, Jan. 2012.
- C. N. Pang, Y. W. Lai, L. T. Campbell, S. C. Chen, D. A. Carter, and M. R. Wilkins, "Transcriptome and network analyses in *Saccharomyces cerevisiae* reveal that amphotericin B and lactoferrin synergy disrupt metal homeostasis and stress response," *Sci. Rep.*, vol. 7, Jan. 12, 2017, Art. no. 40232.
- A. Conesa *et al.*, "A survey of best practices for RNA-seq data analysis," *Genome Biol.*, vol. 17, Jan. 26, 2016, Art. no. 13.
- H. B. TW, and T. Girke, "SystemPipeR: NGS workflow and report generation environment," *BMC Bioinf.*, vol. 17, Sep. 20, 2016, Art. no. 388.
- P. Cliften, "Base calling, read mapping, and coverage analysis," *Clinical Genomics*. Amsterdam, The Netherlands: Elsevier, 2015, pp. 91–107.
- B. Ewing and P. Green, "Base-calling of automated sequencer traces using phred. II. Error probabilities," *Genome Res.*, vol. 8, no. 3, pp. 186–194, Mar. 1998.
- A. Dobin and T. R. Gingeras, "Mapping RNA-seq reads with STAR," *Current Protocol Bioinf.*, vol. 51, pp. 11 14 1–11 14 19, Sep. 2015.
- A. C. Frazee, G. Pertea, A. E. Jaffe, B. Langmead, S. L. Salzberg, and J. T. Leek, "Ballgown bridges the gap between transcriptome assembly and expression analysis," *Nature Biotechnology*, vol. 33, no. 3, pp. 243–246, Mar. 2015.
- M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biol.*, vol. 15, no. 12, 2014, Art. no. 550.
- D. J. McCarthy, Y. Chen, and G. K. Smyth, "Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation," *Nucl. Acids Res.*, vol. 40, no. 10, pp. 4288–4297, May 2012.
- M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: A Bioconductor package for differential expression analysis of digital gene expression data," *Bioinf.*, vol. 26, no. 1, pp. 139–140, Jan. 1, 2010.
- H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Berlin, Germany: Springer, 2016.
- T. Wei and V. Simko, *R package "corrplot": Visualization of a Correlation Matrix*, 2017, (Version 0.84). [Online]. Available: <https://github.com/taiyun/corrplot>
- A. E. Berglund, E. A. Welsh, and S. A. Eschrich, "Characteristics and validation techniques for PCA-based gene-expression signatures," *Int. J. Genomics*, vol. 2017, 2017, Art. no. 2354564.
- S. Lè, J. Josse, and F. J. J. O. S. S. Husson, "FactoMineR: An R package for multivariate analysis," vol. 25, no. 1, pp. 1–18, 2008.
- R. Kolde, *pheatmap: Pretty Heatmaps*, 2019, R package version 1.0.12. [Online]. Available: <https://CRAN.R-project.org/package=pheatmap>
- G. Yu, L. G. Wang, Y. Han, and Q. Y. He, "clusterProfiler: An R package for comparing biological themes among gene clusters," *OMICS*, vol. 16, no. 5, pp. 284–287, May. 2012.
- M. Ashburner *et al.*, "Gene ontology: Tool for the unification of biology. The gene ontology consortium," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May. 2000.
- The Gene Ontology Consortium, "Expansion of the gene ontology knowledgebase and resources," *Nucl. Acids Res.*, vol. 45, no. D1, pp. D331–D338, Jan. 4, 2017.
- M. Kanehisa and S. Goto, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucl. Acids Res.*, vol. 28, no. 1, pp. 27–30, Jan. 1, 2000.
- H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucl. Acids Res.*, vol. 27, no. 1, pp. 29–34, Jan. 1, 1999.
- W. Luo and C. Brouwer, "Pathview: An R/Bioconductor package for pathway-based data integration and visualization," *Bioinf.*, vol. 29, no. 14, pp. 1830–1831, Jul. 2013.
- T. Barrett *et al.*, "NCBI GEO: Archive for functional genomics data sets—update," *Nucl. Acids Res.*, vol. 41, no. Database issue, pp. D991–D995, Jan. 2013.
- R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: NCBI gene expression and hybridization array data repository," *Nucl. Acids Res.*, vol. 30, no. 1, pp. 207–210, Jan. 2002.
- L. A. Martin *et al.*, "Discovery of naturally occurring ESR1 mutations in breast cancer cell lines modelling endocrine resistance," *Nature Commun.*, vol. 8, no. 1, Nov. 2017, Art. no. 1865.
- S. K. Kim *et al.*, "A nineteen gene-based risk score classifier predicts prognosis of colorectal cancer patients," *Mol. Oncol.*, vol. 8, no. 8, pp. 1653–1666, Dec. 2014.
- R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2019. [Online]. Available: <https://www.R-project.org/>
- R. Gentleman, V. Carey, W. Huber, R. Irizarry, and S. Dudoit, *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Berlin, Germany: Springer, 2006.
- R. C. Gentleman *et al.*, "Bioconductor: Open software development for computational biology and bioinformatics," *Genome Biol.* vol. 5, no. 10, 2004, Art. no. R80.

- [41] B. Grüning *et al.*, "Bioconda: Sustainable and comprehensive software distribution for the life sciences," *Nature Methods*, vol. 15, no. 7, pp. 475–476, 2018.
- [42] J. Wu, O. Anczukow, A. R. Krainer, M. Q. Zhang, and C. Zhang, "Olego: Fast and sensitive mapping of spliced mRNA-Seq reads using small seeds," *Nucl. Acids Res.*, vol. 41, no. 10, pp. 5149–5163, May 1, 2013.
- [43] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biol.*, vol. 14, no. 4, Apr. 25, 2013, Art. no. R36.
- [44] C. Trapnell *et al.*, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, vol. 28, no. 5, pp. 511–515, May 2010.
- [45] A. I. Tomescu, A. Kuosmanen, R. Rizzi, and V. Makinen, "A novel min-cost flow method for estimating transcript expression with RNA-Seq," *BMC Bioinf.*, vol. 14, no. Suppl 5, 2013, Art. no. S15.
- [46] A. C. Frazee, G. Pertea, A. E. Jaffe, B. Langmead, S. L. Salzberg, and J. T. Leek, "Flexible isoform-level differential expression analysis with ballgown," *Biorxiv*, p. 003665, 2014.
- [47] J. Goecks, A. Nekrutenko, J. Taylor, and T. Galaxy, "Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biol.*, vol. 11, no. 8, 2010, Art. no. R86.
- [48] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: A tool for building and running workflows of services," *Nucl. Acids Res.*, vol. 34, no. Web Server issue, pp. W729–W732, Jul. 2006.
- [49] J. Koster and S. Rahmann, "Snakemake—a scalable bioinformatics workflow engine," *Bioinf.*, vol. 34, no. 20, Oct. 2018, Art. no. 3600.
- [50] P. D. Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. J. N. B. Notredame, "Nextflow enables reproducible computational workflows," *Nature Biotechnology*, vol. 35, no. 4, pp. 316–319, 2017.
- [51] C. Guzman and I. D'Orso, "CIPHER: A flexible and extensive workflow platform for integrative next-generation sequencing data analysis and genomic regulatory element prediction," *BMC Bioinf.*, vol. 18, no. 1, Aug. 2017, Art. no. 363.
- [52] R. V. J. E. J. Guimera, "bcbio-nextgen: Automated, distributed next-gen sequencing pipeline," *EMBnet. J.*, vol. 17, no. B, 2011, Art. no. 30.
- [53] M. Wolfien *et al.*, "TRAPLINE: A standardized and automated pipeline for RNA sequencing data analysis, evaluation and annotation," *BMC Bioinf.*, vol. 17, Jan. 2016, Art. no. 21.
- [54] D. Wang, "hppRNA—a Snakemake-based handy parameter-free pipeline for RNA-Seq analysis of numerous samples," *Brief Bioinf.*, vol. 19, no. 4, pp. 622–626, Jul. 2018.
- [55] S. Zhao *et al.*, "QuickRNASeq lifts large-scale RNA-seq data analyses to the next level of automation and interactive visualization," *BMC Genomics*, vol. 17, no. 1, 2016, Art. no. 39.
- [56] D. Gaidatzis, A. Lerch, F. Hahne, and M. B. Stadler, "QuasR: Quantification and annotation of short reads in R," *Bioinf.*, vol. 31, no. 7, pp. 1130–1132, Apr. 2015.
- [57] M. J. Steinbaugh *et al.*, "bcbioRNASeq: R package for bcbio RNA-seq analysis," *F1000Research*, vol. 6, 2018.



Kuan-Hao Chao is currently working toward the undergraduate degree in the Department of Electrical Engineering at the National Taiwan University (NTU). His research interests include developing bioinformatics online systems and bioconductor R package.



Yi-Wen Hsiao received the master's degree from the Industrial and Commercial Biotechnology Program, Newcastle University. She is a research assistant in the Bioinformatics and Biostatistics Core Lab, Center of Genomic and Precision Medicine, National Taiwan University.



Yi-Fang Lee received the bachelor's degree in electrical engineering, and MSc degree in bioinformatics from National Taiwan University (NTU), Taiwan. She is currently working toward the Electrical Engineering and Computer Science (EECS) master of engineering degree at UC Berkeley. With interests in both engineering and biology. Her research interests include developing bioinformatics databases and online systems.



Chien-Yueh Lee received the BSc degree in electrical engineering from the National Taipei University of Technology, Taiwan, and the MSc and PhD degrees in biomedical electronics and bioinformatics from National Taiwan University, Taiwan. He is a postdoctoral researcher in the Perelman School of Medicine, University of Pennsylvania, USA. His research interests include bioinformatics, computational biology, genomics, big data analysis, and database.



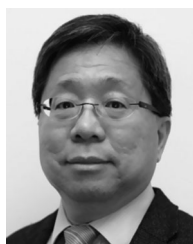
Liang-Chuan Lai received the PhD degree from the University of Illinois at Urbana-Champaign, in 2005. He is an associate professor in the Institute of Biotechnology, National Taiwan University. His main research interests include genomic approaches to explore the molecular mechanism of carcinogenesis.



Mong-Hsun Tsai received the PhD degree from the Institute of Public Health, National Yang-Ming University, in 2001. He is a professor with the Institute of Biotechnology, National Taiwan University. His main research interests include bioinformatics, cell biology, microarray, and radiation biology.



Tzu-Pin Lu received the PhD degree from the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University. He is an associate professor in the Institute of Epidemiology and Preventive Medicine, College of Public Health, National Taiwan University. His major research interests include bioinformatics and computational biology.



Eric Y. Chuang received the ScD in cancer biology from Harvard University, in 1997. He is a professor in the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University. His major research interests include biochip, bioinformatics, and cancer biology.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.