clusterProfiler 4.0: A universal enrichment tool for interpreting omics data

Tianzhi Wu,1,5 Erqiang Hu,1,5 Shuangbin Xu,1 Meijun Chen,1 Pingfan Guo,1 Zehan Dai,1 Tingze Feng,1 Lang Zhou,1 Wenli Tang,1 Li Zhan,1 Xiaocong Fu,1 Shanshan Liu,1 Xiaochen Bo,2,* and Guangchuang Yu1,3,4,*

*Correspondence: boxc@bmi.ac.cn (X.B.); gcyu1@smu.edu.cn (G.Y.)

Received: May 8, 2021; Accepted: June 29, 2021; Published Online: July 1, 2021; https://doi.org/10.1016/j.xinn.2021.100141

© 2021 The Author(s). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Public summary
- clusterProfiler supports exploring functional characteristics of both coding and non-coding genomics data for thousands of species with up-to-date gene annotation
- It provides a universal interface for gene functional annotation from a variety of sources and thus can be applied in diverse scenarios
- It provides a tidy interface to access, manipulate, and visualize enrichment results to help users achieve efficient data interpretation
- Datasets obtained from multiple treatments and time points can be analyzed and compared in a single run, easily revealing functional consensus and differences among distinct conditions
Functional enrichment analysis is pivotal for interpreting high-throughput omics data in life science. It is crucial for this type of tool to use the latest annotation databases for as many organisms as possible. To meet these requirements, we present here an updated version of our popular Bioconductor package, clusterProfiler 4.0. This package has been enhanced considerably compared with its original version published 9 years ago. The new version provides a universal interface for functional enrichment analysis in thousands of organisms based on internally supported ontologies and pathways as well as annotation data provided by users or derived from online databases. It also extends the dplyr and ggplot2 packages to offer tidy interfaces for data operation and visualization. Other new features include gene set enrichment analysis and comparison of enrichment results from multiple gene lists. We anticipate that clusterProfiler 4.0 will be applied to a wide range of scenarios across diverse organisms.

Keywords: clusterProfiler; biological knowledge mining; functional analysis; enrichment analysis; visualization

INTRODUCTION

Functional enrichment analysis is one of the most widely used techniques for interpreting gene lists or genome-wide regions of interest (ROIs) derived from various high-throughput studies. Although many tools have been developed for gene-centric or epigenomic enrichment analysis, most are designed for model organisms or specific domains (e.g., fungi, plants) embedded with particular annotations such as Gene Ontology (GO) and the Kyoto Encyclopedia of Genes and Genomes (KEGG). Non-model organisms and functional annotations other than GO and KEGG are poorly supported. Moreover, the increasing concerns for the quality of gene annotation have raised an alarm in biomedical research. Because annotation databases have diverse or irregular update periods, many tools may fail to update the corresponding information in time. A previous study reported that about 42% of the tools were outdated by more than 5 years and that functional significance was severely underestimated, with only 26% of biological processes or pathways captured in comparison with those employing up-to-date annotation. Such negative impacts of outdated annotation can be propagated for years and can hinder follow-up studies. Reanalyzing the GTEx consortium using clusterProfiler uncovered a large number of new pathways, which were missed in the analysis using out-of-date annotation (https://github.com/GuangchuangYu/enrichment4GTEx_clusterProfiler), and new hypotheses were generated based on these new pathways.

The clusterProfiler library was first published in 2012 and designed to perform over-representation analysis (ORA) using GO and KEGG for several model organisms and to compare functional profiles of various conditions on one level (e.g., different treatment groups). Since then, clusterProfiler has matured substantially and currently supports several ontology and pathway annotations, thousands of species with up-to-date gene annotation, users’ annotation data for novel species, and emerging new annotations. Both ORA and gene set enrichment analysis (GSEA) are supported. The comparison utility is extended to support a complex experimental design that allows comparison of functional profiles of various conditions on different levels. The clusterProfiler library has many unique features, including a tidy interface that can manipulate the enrichment result and directly support the visualization of the enrichment result using ggplot2 (Tables 1 and S2). Moreover, we have developed several packages to complement its functionalities, including ChIPseeker to connect functional analysis with genomic ROIs, GOSemSim to remove redundant GO terms, and enrichplot to visualize the enrichment results. These complementary packages enable clusterProfiler to stand out among other tools. The clusterProfiler library is one of the most popular Bioconductor packages. It has been incorporated in more than 30 CRAN and Bioconductor packages (Table S1), several pipelines (e.g., The Cancer Genome Atlas [TCGA] Workflow and ViralInk), and online platforms (e.g., NASQAR and ABIoTrans).

RESULTS

Gene ontology

The clusterProfiler package provides the enrichGO and gseGO functions for ORA and GSEA using GO. Instead of providing species-specific GO annotation, clusterProfiler relies on genome-wide annotation packages (OrgDb) released by the Bioconductor project. There are 20 OrgDb packages available in Bioconductor for different species, such as human, mouse, fly, yeast, and worm. These packages are updated biannually. GO annotation for non-model organisms can be queried online via the AnnotationHub package, which provides web services for accessing genome-wide annotations from various data providers (e.g., UCSC, Ensembl, NCBI, STRING, and GENCODE). With the efforts from the Bioconductor community to maintain up-to-date GO annotation for model and non-model organisms, clusterProfiler supports GO analysis on more species compared with other tools. Moreover, a data frame of GO annotation (e.g., retrieve data from the BiomaRt or UniProt database using taxonomic ID) can be used to construct an OrgDb using the AnnotationForge package or directly through the universal interface for enrichment analysis.

GO terms are organized as a directed acyclic graph, in which a directed edge denotes a parent-child semantic relationship. A parent term might be significantly enriched only because it contains all the genes of a significantly over-represented child term. Consequently, the list of enriched GO terms is often too long and contains redundant terms, which hinders effective interpretation. Therefore, clusterProfiler integrates a simplify function to eliminate such redundant GO terms. This function employs the GOSemSim package to calculate semantic similarities among enriched GO terms using multiple methods based on information content or graph structure.
Highly similar GO terms (e.g., >0.7) will be removed by applying the simplify function to retain a representative term (e.g., the most significant term). The following example shows an ORA on Biological Process (BP) to identify significant BP terms associated with the differentially expressed genes (DEGs). The geneList dataset, which contains fold change of gene expression levels between breast tumor and normal samples and is provided by the DOSE package, was used in this example. The DEGs were identified by a criterion of fold change >2. As demonstrated in Figure 1A, the top 30 enriched terms are highly connected, and it seems that the DEGs are associated with a single functional module. Visualizing top enriched terms is a common approach to present and interpret the enrichment result. However, the top results are dominated by a large number of highly similar terms. After removing redundant terms, the result reveals a more global view with several different functional modules (Figure 1B). This feature simplifies the enrichment results, assists in interpretation, and avoids the annotation/interpretation bias.18

Table 1. Major clusterProfiler functions

| Function       | Description                                      |
|----------------|--------------------------------------------------|
| enrichGO       | ORA using GO                                     |
| enrichKEGG     | ORA using KEGG pathway                           |
| enrichMKEGG    | ORA using KEGG module                            |
| enrichWP       | ORA using WikiPathways                           |
| enricher       | general interface for ORA                        |
| gseGO          | GSEA using GO                                    |
| gseKEGG        | GSEA using KEGG pathway                          |
| gseMKEGG       | GSEA using KEGG module                           |
| gseWP          | GSEA using WikiPathways                          |
| GSEA           | general interface for GSEA                       |
| compareCluster | compare functional profiles for genes obtained from different conditions |
| merge_result   | merge enrichment results for comparison          |
| read.gmt       | parse gene set file in GMT format                |
| read.gmt.wp    | parse WikiPathways GMT file                      |
| download_KEGG  | download the latest version of the KEGG pathway/module |
| get_wp_organism| list supported organisms of WikiPathways         |
| bitr           | biological ID translator using OrgDb             |
| bitr_kegg      | biological ID translator using the KEGG database  |
| setReadable    | convert IDs in enrichment result to human-readable gene symbols using OrgDb |
| go2ont         | convert GO ID to corresponding ontology (BP, CC, MF) |
| go2term        | convert GO ID to a descriptive term              |
| ko2name        | convert KO ID to a descriptive name              |
| buildGOmap     | infer GO indirect annotation from direct annotation |
| browseKEGG     | open specific KEGG pathway in a web browser with genes highlighted |
| dropGO         | drop GO terms of specific level or a specific terms (mostly too general) from enrichment result |
| gofilter       | restrict enrichment result at a specific GO level |
| geneInCategory | extract input genes (for ORA) or core enriched genes (for GSEA) that belong to a specific functional category |
| simplify       | remove redundant GO terms from enrichment result |
| arrange        | order enrichment result by the values of selected variables |
| filter         | subset enrichment result that satisfies user conditions |
| group_by       | group enrichment results by selected variable    |
| mutate         | add new variable to enrichment result            |
| select         | select variables in enrichment result            |
| summarise      | create summary statistics from enrichment result |
KEGG is an encyclopedia of genes and genomes. Molecular functions are represented by networks of interactions and reactions mainly in the form of KEGG pathways and modules. A KEGG module is a collection of manually defined function units. In some situations, KEGG modules have a more straightforward interpretation. Both KEGG pathways and KEGG modules are supported by clusterProfiler. Many software tools that support KEGG analysis have stopped updating since July 2011 when KEGG initiated an academic subscription model for FTP downloading. These tools use relatively old KEGG data, and the result might be inaccurate and misleading. Fortunately, the KEGG web resource is freely available. The clusterProfiler package does not pack any KEGG data. Instead, it queries the latest online KEGG database through web API to perform functional analysis. The advantage of this feature is obvious: it allows clusterProfiler to use up-to-date data and support all the species that have KEGG annotation (more than 6,000 species are listed in http://www.genome.jp/kegg/catalog/org_list.html). Moreover, clusterProfiler supports the KEGG Orthology database and can be used to perform functional characterization of the microbiomes.

In the following example, GSEA was performed with KEGG pathway. Figure 2A shows the plotting of GSEA enrichment results to visualize the top five perturbed pathways, i.e., the top five highest absolute values of the normalized enrichment score (NES). The NES indicates the shift of genes belonging to a certain pathway toward either end of the ranked list and represents pathway activation or suppression. To further explore the pathway crosstalk effects, we visualized gene expression distribution of core enrichment genes using an UpSet plot (Figure 2B). The result shows that the expression values of genes in the intersection of cell-cycle and DNA-replication pathways are higher than those uniquely belonging to either of the two pathways. These overlapping genes are mainly minichromosome maintenance (MCM) genes, which can potentially serve as biomarkers for tumor diagnosis. The intersection of the interleukin-17 (IL-17) signaling pathway and the proteasome pathway is only associated with one gene, interferon-γ (IFN-γ). The IL-17 signaling pathway induces an inflammatory response, while IFN-γ regulates proteasome formation. These effects ultimately reshape the tumor microenvironment.

Universal interface for biomedical gene sets

With the advancement of the sequencing technology, the investigation into functions for transcriptomes from non-model organisms is increasingly demanded. However, most tools in this field are designed for GO and KEGG analyses with support limited to one or several model organisms. Besides, there are increasingly more biological knowledge databases available for exploring functional characteristics from different perspectives, such as Disease Ontology, Reactome Pathway, Medical Subject Headings, and WikiPathway. There is an urgent need for integration and support of these databases. To address these issues, clusterProfiler provides two general functions, enricher and GSEA for ORA and GSEA, with user-provided gene annotations. These two functions allow the application of all ontologies or pathways curated in diverse databases as the background in customized analyses. Therefore, users could easily import external annotations (e.g., electronic annotations using Blast2GO and KAAS for GO and KEGG annotations, respectively) for newly sequenced species. Moreover, it is convenient...
to perform functional analysis using up-to-date annotations from all popular databases, such as InterPro, Clusters of Orthologous Groups, and Mouse Phenotype Ontology, to name a few, without waiting for the updates of other tools. It would be suitable for the timely analysis of gene sets with emerging interests, such as human cell markers and COVID-19-related gene sets.

The gene set annotation required by enricher and GSEA is a two-column data frame with one column representing gene set names (ID or descriptive name) and the other showing the corresponding genes. The gene matrix transposed (GMT) format is widely used to distribute gene set annotations. There are many gene set libraries available online (e.g., https://maayanlab.cloud/Enrichr/stats), including MSigDB (Molecular Signatures Database), Disease Signatures, and CCLE (Cancer Cell Line Encyclopedia). To enable the utilization of these gene sets in clusterProfiler as the background annotation to explore the underlying biological mechanisms, clusterProfiler provides a parser function, read.gmt, to import GMT files that can be directly passed to the enricher and GSEA functions. In the following example, we used the GSEA function to perform gene set enrichment analysis using WikiPathways (Figure 5B). The annotation data were parsed by using read.gmt.wp, which is a customized version of read.gmt for importing GMT files from WikiPathways.

Functional interpretation of genomic ROIs

With the increasing availability of genomic sequences, non-coding genomic regions (e.g., cis-regulatory elements, non-coding RNAs, and transposons) have posed a demanding challenge to exploration of their roles in various biological processes. Unlike coding genes, non-coding genomic regions are typically not well functionally annotated. Analyzing biological functions of the proximal genes is a common strategy in research on the biological meaning of a set of non-coding genomic regions. Software tools, such as the Genomic Regions Enrichment of Annotations Tool (GREAT), are implemented to follow this strategy. However, these tools only support a limited number of species. For example, GREAT is designed for human and mouse only. In addition, many tools only take the host or nearest genes into consideration but ignore long-distance regulations. Our in-house developed package, ChIPseeker, is originally designed for chromatin immunoprecipitation (ChIP) peak annotation, comparison, and visualization and has been employed to analyze genome-wide ROIs, such as open chromatin regions obtained by DNase-seq and ATAC-seq. To facilitate biological interpretation of genome-wide regions, we implemented a function, seq2gene, in ChIPseeker to associate genomic regions with coding genes through many-to-many mapping. It automatically maps genomic regions to host genes (either located in exon or intron), proximal genes (located in the promoter region), and flanking genes (located upstream and downstream within user-specified distance). The seq2gene function supports a wide variety of species if a genomic annotation, such as the TxDb (UCSC-based) or Ensembl-based object, is available. After mapping genomic regions to coding genes, clusterProfiler can be employed to perform functional enrichment analysis of the coding genes to assign biological meanings to the set of genomic regions. The combination of ChIPseeker and clusterProfiler allows more biological ontology or pathway databases to be utilized to explore functions of genomic regions for a wide variety of species.

```r
library(ChIPseeker)

## the file can be downloaded using 'downloadGSMbedFiles'
file <- "GSM1295076_CBX6_BF_ChipSeq_mergedReps_peaks.bed.gz"

gr <- readPeakFile(file)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)

TxDb <- TxDb.Hsapiens.UCSC.hg19.knownGene
genes <- seq2gene(gr, tssRegion=c(-1000, 1000),
                   flankDistance = 3000, TxDb)
```
A dataset of ChIP-seq with antibody against CBX6 (GEO: GSM1295076) was used in the above example. The genomic binding regions were mapped to coding genes using the seq2gene function with UCSC genomic annotation. The Entrez gene IDs were converted into gene symbols using the bitr function implemented in clusterProfiler. To identify and characterize transcript cofactors, we performed functional enrichment analysis using the ENCODE and ChEA transcript factor gene sets. The result was visualized as a category-gene network (Figure 3), which showed that genes associated with CBX6 (obtained by the seq2gene function) significantly overlap with genes regulated by POU5F1, TRIM28, SUZ12, and EZH2. OCT4 (POU5F1) and KAP1 (TRIM28) are core components of PRC2 and negatively regulate CBX6.\(^37\) These pieces of evidence support the effectiveness of these analyses including the mapping of genomic ROIs to coding genes and functional enrichment, which suggest that this method can be used to identify unknown cofactors (Figure 3) and characterize functions of genomic regions.

### Comparison among different conditions

The clusterProfiler library is designed to allow the comparison of functional enrichment results from multiple experimental conditions or multiple time points. With an input of a collection of gene lists, the compareCluster function applies a function (e.g., enricher) with user settings to perform functional enrichment analysis for each of the gene lists and aggregates the results into a single object. Thus, enrichment results of multiple groups are easily explored and plotted together for comparison with a user-friendly interface. Comparing functional profiles can reveal functional consensus and differences among different experiments and helps in identifying differential functional modules in omics datasets. In the updated version, compareCluster provides a new interface supporting a formula that is widely used in R for specifying statistical models; this allows more complicated experimental designs (e.g., time-course experiment with different treatments). With the infrastructure of clusterProfiler to support a wide range of ontology and pathway annotations and multiple organisms, the comparison can be applied to many circumstances.

The dataset, DE_GSE8057, was derived from the GEO: GSE8057 dataset in the GEO database. The GSE8057 dataset contains expression data from...
ovarian cancer cells at multiple time points (0, 2, 6, and 24 h) and under two treatment conditions (cisplatin and oxaliplatin). The DE_GSE8057 dataset contains DEGs obtained from different treatments and time points versus control samples. Eight groups of DEG lists (specified by the formula Gene ~ Time + Treatment) were analyzed simultaneously using compareCluster with WikiPathways. The result (Figure 4) indicates that the two drugs have distinct effects at the beginning but consistent effects in the later stages. Several pathways including DNA damage and cell-cycle progression were perturbed by either cisplatin or oxaliplatin drug exposure. The finding is consistent with the discovery obtained by data-driven modeling.

The dataset contains DEGs obtained from different treatments and time points versus control samples. Eight groups of DEG lists (specified by the formula Gene ~ Time + Treatment) were analyzed simultaneously using compareCluster with WikiPathways. The result (Figure 4) indicates that the two drugs have distinct effects at the beginning but consistent effects in the later stages. Several pathways including DNA damage and cell-cycle progression were perturbed by either cisplatin or oxaliplatin drug exposure. The finding is consistent with the discovery obtained by data-driven modeling.

**Data frame interface for accessing enriched results**

The outputs of ORA and GSEA are enrichResult and gseaResult objects, respectively, while the output of compareCluster is a compareClusterResult object. These S4 objects contain input data, analysis settings, and enriched results, which allow more informative data to be available for downstream interpretation and visualization. To enable easy access to the enriched result, clusterProfiler implements as.data.frame methods to convert the S4 objects to data frames that can be easily exported as CSV files. In addition, clusterProfiler provides a data frame interface that mimics data frame operations to access rows, columns, and sub-sets of rows and columns from the S4 objects of the enriched result. Users can use head and tail to print part of the result. The nrow, ncol, and dim methods are also supported to access basic information such as how many pathways are enriched.

```r
data(DE_GSE8057)
xx <- compareCluster(Gene=time+treatment,
data=DE_GSE8057, fun=enricher,
TERM2GENE=wp[,c("wpid", "gene")],
TERM2NAME=wp[,c("wpid", "name")])
```

**Figure 4. Comparing functional profiles among different levels of conditions**

The compareCluster function performed enrichment analysis simultaneously for eight lists of DEGs. The results were visualized as a dot plot with an x axis representing one level of conditions (time course) and a facet panel indicating another level of conditions (drug treatments).

```r
head(xx, 2)
```

| ID          | Description                      | GeneRatio | BgRatio | pvalue        |
|-------------|----------------------------------|-----------|---------|---------------|
| GO:0140014  | GO:0140014 mitotic nuclear division | 34/194    | 286/18866 | 2.171838e-26  |
| GO:0000280  | GO:0000280 nuclear division       | 36/194    | 428/18866 | 1.099719e-22  |
| p.adjust   |                                  |           |         |               |
| GO:0140014  | 6.700119e-23                |           |         |               |
| GO:0000280  | 1.696316e-19                 |           |         |               |
| geneID      |                                  |           |         |               |
| GO:0140014  | CDCA8/CDC20/KIF23/CENPE/NYBL2/CCHB2/ND80/NAP1/DP/LGAP5/UBE2C/NUSAP1/TPX2/TACC3/NEK2/UBE2S/CDK1/MAD2L1/KIF18A/CDT1/BIRC5/KIF11/TTK/NAPC/AURKB/CHEX1/TRIP13/PKC1/KIFC1/KIF18B/AURKA/CNN1/KIF4A/PTTG1/BMP4 |
| GO:0000280  | CDCA8/CDC20/KIF23/CENPE/NYBL2/CCHB2/ND80/TOP2A/NAP1/DP/LGAP5/UBE2C/NUSAP1/TPX2/TACC3/NEK2/UBE2S/CDK1/MAD2L1/KIF18A/CDT1/BIRC5/KIF11/TTK/NAPC/AURKB/CHEX1/TRIP13/PKC1/KIFC1/KIF18B/AURKA/CNN1/KIF4A/PTTG1/BMP4 |
| Count       |                                  |           |         |               |
| GO:0140014  | 6.700119e-23                |           |         |               |
| GO:0000280  | 1.696316e-19                 |           |         |               |
```

The [ and $ operators for subsetting are also supported. We redefined the ] operator to help users access which genes are annotated by a selected
For GSEA output, the `%` operator will return core enriched genes (i.e., genes in the leading edge) of the selected gene set.

```r
ego3 <- mutate(ego, richFactor = Count / as.numeric(sub("/\d+", "", BgRatio)))
```

The following example uses the GSEA enrichment result generated in the previous session. The result was sorted by absolute values of NESs using the `arrange` verb. NES is an indicator to interpret the degree of enrichment. A positive NES indicates that members of the gene set tend to appear at the top of the rank (pathway activation), and a negative NES indicates the opposite circumstance (pathway suppression). We used the `group_by` verb to group the result based on the sign of NES, and the `slice` verb was used to extract the first five enriched pathways for each group (i.e., five activated pathways that have the largest NES values and five suppressed pathways that have the smallest NES values). These verbs return the same object type as their input and do not affect downstream analysis and visualization.

```r
ewp2 <- arrange(ewp, desc(abs(NES))) %>%
  group_by(sign(NES)) %>%
  slice(1:5)
```

### Tidy interface for data operation

To facilitate data manipulation and exploration of the enrichment result, clusterProfiler extends the `dplyr` verbs to support `enrichResult`, `gseaResult`, and `compareClusterResult` objects. Following the concept of tidiness, these verbs provide robust and standardized operations for data transformation and can be assembled into a workflow using the pipe operator (%>%). This allows users to explore the results effectively and develop reproducible and human-readable pipelines. For example, it allows the filtering of enriched results using different criteria (e.g., adjusted p values less than 0.001, and the number of input genes annotated to the enriched term should be greater than 10).

```r
dim(ego)
## [1] 197 9
ego2 <- filter(ego, p.adjust < 0.001, Count > 10)
dim(ego2)
## [1] 44 9
```

For ORA results, clusterProfiler provides `geneRatio` (ratio of input genes that are annotated in a term) and `BgRatio` (ratio of all genes that are annotated in this term). However, other concepts are widely used to help in interpreting enrichment results, such as the rich factor and fold enrichment. A rich factor is defined as the ratio of input genes (e.g., DEGs) that are annotated in a term to all genes that are annotated in this term. The fold enrichment is defined as the ratio of the frequency of input genes annotated in a term to the frequency of all genes annotated to that term, and it is easy to calculate by dividing `geneRatio` by `BgRatio`. Here, as an example, we used the `mutate` verb to create a new column of `richFactor` based on information available in the clusterProfiler output.

```r
ego3 <- mutate(ego, richFactor = Count / as.numeric(sub("/\d+", "", BgRatio)))
```

### Visualization using Ggplot2

The enrichplot package is originally derived from DOSE and clusterProfiler packages and serves as a de facto visualization tool for visualizing enrichment results for outputs from clusterProfiler as well as DOSE, ReactomePA, and meshes. These methods allow users without programming skills to generate effective visualization to explore and interpret results. All the visualization methods implemented are based on `ggplot2`, which allows customization using the grammar of graphics. Moreover, we also extend ggplot2 to support enrichment results so that users can use the ggplot2 syntax directly to visualize enrichment results. The following example demonstrates the application of ggplot2 grammar of graphics to visualize the GO enrichment result (ORA) as a lollipop chart using the rich factor that was generated in the previous session using the `dplyr` verbs (Figure 5A).
The Innovation

Traditional pathway enrichment analysis has been an essential step toward identifying biological themes that are most characteristic of high-throughput sequencing data. The clusterProfiler library provides a set of functions to unveil biological functions and pathways. Compared with many other tools that do not update background annotation databases in timely fashion and only support a limited number of organisms, clusterProfiler uses up-to-date biological knowledge of genes and biological processes (GO and KEGG) and supports thousands of organisms. In addition, clusterProfiler provides a universal interface for functional analysis with user-provided annotations. This creates the possibility to apply clusterProfiler on functional characterization of different types of data with different biological knowledge. The tidy interface provided in clusterProfiler harmonizes data structures and workflows and makes it easier for the community to develop modular manipulation, visualization, and analysis methods to supplement the existing ecosystem. clusterProfiler has already been integrated into more than 30 packages to perform functional analysis on data obtained using different techniques, including ATAC-seq, multi-region sequencing (MRS), CRISPR/Cas9 screens, and mass spectrometry (Table S1). The clusterProfiler package can be easily integrated into analysis pipelines. For example, the Gene Ontology Meta Annotator for Plants (GOMAP) is optimized for GO annotation of large, repetitive plant genomes. Users can develop a pipeline to combine GOMAP with

### DISCUSSION AND CONCLUSIONS

Pathway enrichment analysis is an essential step toward identifying biological themes that are most characteristic of high-throughput sequencing data. The clusterProfiler library provides a set of functions to unveil biological functions and pathways. Compared with many other tools that do not update background annotation databases in timely fashion and only support a limited number of organisms, clusterProfiler uses up-to-date biological knowledge of genes and biological processes (GO and KEGG) and supports thousands of organisms. In addition, clusterProfiler provides a universal interface for functional analysis with user-provided annotations. This creates the possibility to apply clusterProfiler on functional characterization of different types of data with different biological knowledge. The tidy interface provided in clusterProfiler harmonizes data structures and workflows and makes it easier for the community to develop modular manipulation, visualization, and analysis methods to supplement the existing ecosystem. clusterProfiler has already been integrated into more than 30 packages to perform functional analysis on data obtained using different techniques, including ATAC-seq, multi-region sequencing (MRS), CRISPR/Cas9 screens, and mass spectrometry (Table S1). The clusterProfiler package can be easily integrated into analysis pipelines. For example, the Gene Ontology Meta Annotator for Plants (GOMAP) is optimized for GO annotation of large, repetitive plant genomes. Users can develop a pipeline to combine GOMAP with

---

**Figure 5. Visualizing enrichment results using ggplot2** A lollipop chart to visualize the rich factors from ORA (A) and a bar chart to visualize normalized enrichment scores from GSEA (B).

```r
theme_dose(12) +
xlab("Rich Factor") +
lab(NULL) +
ggtitle("Biological Processes")

ggplot(ewp2, showCategory=10, 
aes(NES, fct_reorder(Description, NES), 
fill=qvalues)) +
geom_col() +
scale_fill_gradientn(colours=c("#b3eebe", 
"#46bac2", "#371ea3"), 
guide=guide_colorbar(reverse=TRUE)) +
theme_dose(12) +
 labs("Normalized Enrichment Score") +
ylab(NULL) +
ggtitle("WikiPathways")
```

### Package interoperability

The clusterProfiler package is a versatile tool for enrichment analysis. It is developed within the Bioconductor ecosystem and has become an essential part of this ecosystem. Currently there are more than 30 R packages that rely on clusterProfiler to perform functional analysis for different topics, especially for cancer research. GO analysis relies on GO annotation maintained by the community, and the enrichment analysis for genomic regions relies on genomic annotation maintained by UCSC and Ensembl. There are R packages that contain gene set annotation (e.g., msigdbR) and R client libraries for accessing pathway data (e.g., rWikiPathways). These data can be used directly as background annotation in clusterProfiler through the universal interface to characterize the functional profile of omics data. The ORA algorithm is implemented in the DOSE package developed in-house, and the GSEA algorithm is implemented in DOSE and fgsea packages.

Our team has developed several packages to complement the functionality of clusterProfiler. ChIPseeker bridges the genomic region with functional enrichment by annotating the genomic region to associated genes. GOsimSim provides more than five methods for measuring semantic similarity. It allows removal of redundant terms using semantic similarities among GO terms and allows enrichment results to be visualized in semantic space so that similar terms cluster together. The DOSE package supports functional enrichment from the disease perspective, including disease ontology, the network of cancer genes, and disease gene network. The ReactomePA and meshes packages support functional analysis using Reactome Pathways and Medical Subject Headings, respectively. DOSE, ReactomePA, and meshes are developed within the framework of clusterProfiler, and the enrichment analysis functions provided in these packages can be used in compareCluster for the comparison of functional profiles under various conditions and at different time points. The enrichplot package provides several visualization methods to generate publication-quality figures to help users interpret the results (Figures 1, 2, 3, and 4; supplemental information). This package suite provides a comprehensive set of tools for mining biological knowledge to elucidate and interpret molecular mechanisms (Figure 6).
Figure 6. A package suite for mining biological knowledge clusterProfiler is an essential core for functional analysis, the functionalities of which are enhanced by several companion packages.

clusterProfi ler to characterize the functionality of sequencing data from plant species, including wheat, maize, soy, and rice. The clusterProfi ler library has been incorporated into different pipelines, such as TCGA Workflow, RNASeqR, and MASeCKFluLe.

clusterProfi ler 4.0 contains several new features, including the tidy interface and the compatibility of using ggplot2 for visualization. There is no API change for functional enrichment analyses, and this version is fully compatible with downstream packages (Table S1). After long-term maintenance, clusterProfi ler is mature and unlikely to introduce significant API changes in future development. In the event of an API change, we will maintain backward compatibility for at least 1 year and provide a warning message about the change. The clusterProfi ler library is freely available at https://www.bioconductor.org/packages/clust erProfi ler. The developer version of clusterProfi ler is hosted on GitHub (https://github.com/YuLab-SMU/clusterProfi ler), with many active users. A complete reference of the package suite (Figure 6) is available in the online book, https://yulab-smu. SMU/clusterPro.

REFERENCES

1. Dozmorov, M. G. (2017). Epigenomic annotation-based interpretation of genomic data: from enrichment analysis to machine learning. Bioinformatics 33, 3233–3330.
2. Priebe, S., Kreisel, C., Horn, F., et al. (2015). FungiFun2: a comprehensive online resource for systematic analysis of gene lists from fungal species. Bioinformatics 31, 445–446.
3. Yi, X., Du, Z., and Su, Z. (2013). PlantGSEA: a gene set enrichment analysis toolkit for plant community. Nucleic Acids Res. 41, W98–W103.
4. Nam, D., and Kim, S.-Y. (2008). Gene-set approach for expression pattern analysis. Brief. Bioinform. 9, 189–197.
5. Wadi, L., Meyer, M., Weiser, J., et al. (2016). Impact of outdated gene annotations on pathway enrichment analysis. Nat. Methods 13, 705–706.
6. Méél, M., Ferreira, P. G., Reverter, F., et al. (2015). The human transcriptome across tissues and individuals. Science 348, 660–665.
7. Yu, G., Wang, L.-G., Han, Y., et al. (2012). clusterProfi ler: an R Package for comparing biological themes among gene clusters. OMICS J. Integr. Biol. 16, 284–287.
8. Boyle, E. I., Weng, S., Gollub, J., et al. (2004). GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics 20, 3710–3715.
9. Subramanian, A., Tamayo, P., Mootha, V. K., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. U S A 102, 15545–15550.
10. Yu, G., Wang, L.-G., and He, Q.-Y. (2015). ChIPseeker: an R/Bioconductor package for Chip peak annotation, comparison and visualization. Bioinformatics 31, 2382–2383.
11. Yu, G. (2020). Gene ontology semantic similarity analysis using GOSemSim. Methods Mol. Biol. Clifton NJ 2117, 207–215.
12. Silva, T. C., Colaprico, A., Olsen, C., et al. (2016). TCGA Workflow: analyze cancer genomics and epigenomics data using Bioconductor packages. F1000Research 5, 1542.
13. Trevel, A., Bolar, B., Sudhakar, P., et al. (2021). ViralLink: an integrated workflow to investigate the effect of SARS-CoV-2 on intracellu lar signaling and regulatory pathways. PLOS Comput. Biol. 17, e1009686.
14. Yousif, A., Drou, N., Rowe, J., et al. (2020). NASQAR: a web-based platform for high-throughput sequencing data analysis and visualization. BMC Bioinformatics 21, 267.
15. Zou, Y., Bui, T. T., and Selvarajo, K. (2019). ABioTrans: a biostatistical tool for transcriptomics analysis. Front. Genet. 10, 499.
16. The Gene Ontology Consortium (2019). The gene ontology resource: 20 years and still GOing strong. Nucleic Acids Res. 47, D330–D338.
17. Yu, G., Li, F., Qin, Y., et al. (2010). GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics 26, 976–978.
18. Haynes, W. A., Tomczak, A., and Khati, P. (2018). Gene annotation bias impedes biomedical research. Sci. Rep. 8, 1362.
19. Kanehisa, M., Furumichi, M., Tanabe, M., et al. (2017). KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 45, D333–D336.
20. Zolti, A., Green, S. J., Sela, N., et al. (2020). The microbiome as a biosensor: functional profiles elucidate hidden stress in hosts. Microbiome 8, 71.
21. Allison, M. R., Hunt, T., and Forbes, S. J. (2002). Minichromosome maintenance (MCM) proteins may be pre-cancer markers. Gut 50, 290–291.
22. Miossec, P. (2021). Local and systemic effects of IL-17 in joint inflammation: a historical perspective from discovery to targeting. Cell. Mol. Immunol. 18, 860–865.
23. Heinik, S., Ludwig, D., Klotetzki, P.-M., et al. (2005). IFN-γ-induced immune adaptation of the proteasome system is an accelerated and transient response. Proc. Natl. Acad. Sci. U S A 102, 9241–9246.
24. Yu, G., Wang, L.-G., Yan, G.-R., et al. (2015). DOSE: an R/Bioconductor package for disease ontology semantic and enrichment analysis. Bioinformatics 31, 608–609.
25. Yu, G., and He, Q.-Y. (2016). ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization. Mol. Biosyst. 12, 477–479.

26. Yu, G. (2018). Using meshes for MeSH term enrichment and semantic analyses. Bioinformatics 34, 3766–3767.

27. Martens, M., Ammar, A., Ruutu, A., et al. (2021). WikiPathways: connecting communities. Nucleic Acids Res. 49, D613–D621.

28. Conesa, A., Götz, S., García-Gómez, J.M., et al. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. Bioinformatics 21, 3674–3676.

29. Moriya, Y., Itoh, M., Okuda, S., et al. (2007). KAAS: an automatic genome annotation and pathway reconstruction server. Nucleic Acids Res. 35, W182–W185.

30. Zhang, X., Lan, Y., Xu, J., et al. (2019). CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Res. 47, D721–D728.

31. McLean, C.Y., Bristor, D., Hiller, M., et al. (2010). GREAT improves functional interpretation of cis-regulatory regions. Nat. Biotechnol. 28, 495–501.

32. Liu, Y., Fu, L., Kaufmann, K., et al. (2019). A practical guide for DNase-seq data analysis: from data management to common applications. Brief. Bioinform 20, 1865–1877.

33. Yin, S., Lu, K., Tan, T., et al. (2020). Transcriptomic and open chromatin atlas of high-resolution anatomical regions in the rhesus macaque brain. Nat. Commun. 11, 474.

34. Oliviero, G., Munawar, N., Watson, A., et al. (2015). The variant Polycomb Repressor Complex 1 component PCGF1 interacts with a pluripotency sub-network that includes DPPA4, a regulator of embryogenesis. Sci. Rep. 5, 18388.

35. Cheng, B., Ren, X., and Keppola, T.K. (2014). KAPI1 represses differentiation-inducible genes in embryonic stem cells through cooperative binding with PRC1 and derepresses pluripotency-associated genes. Mol. Cell. Biol. 34, 2075–2091.

36. Santanach, A., Blanco, E., Jiang, H., et al. (2017). The Polycomb group protein CBX6 is an essential regulator of embryonic stem cell identity. Nat. Commun. 8, 1235.

37. Deng, H., Gao, X., Gong, L., et al. (2019). CBX6 is negatively regulated by EZH2 and plays a potential tumour suppressor role in breast cancer. Sci. Rep. 9, 197.

38. Brun, Y.F., Varma, R., Hector, S.M., et al. (2008). Simultaneous modeling of concentration-effect and time-course patterns in gene expression data from microarrays. Cancer Genomics Proteomics 5, 43–53.

39. Korotkevich, G., Sukhov, V., Budin, N., et al. (2021). Fast gene set enrichment analysis. bioRxiv. https://doi.org/10.1101/860012.

40. Wimalanathan, K., and Lawrence-Dill, C.J. (2021). Gene ontology Meta annotator for plants (GOMAP). bioRxiv. https://doi.org/10.1101/859988.

41. Collado-Torres, L., Nellore, A., and Jaffe, A.E. (2017). Recount workflow: accessing over 70,000 human RNA-seq samples with Bioconductor. F1000Research. 6, 1558.

42. Chao, K.-H., Hsiao, Y.-W., Lee, Y.-F., et al. (2019). RNASeqR: an R package for automated two-group RNA-Seq analysis workflow. IEEE/ACM Trans. Comput. Biol. Bioinform. https://doi.org/10.1109/TCBB.2019.2956708.

43. Wang, B., Wang, M., Zhang, W., et al. (2019). Integrative analysis of pooled CRISPR genetic screens using MAGeCKFlute. Nat. Protoc. 14, 756–790.

ACKNOWLEDGMENTS
This work was supported by a startup fund from Southern Medical University.

AUTHOR CONTRIBUTIONS
G.Y. and X.B. planned the study, analyzed and interpreted the data, and drafted the manuscript. T.W. and E.H. analyzed and interpreted the data, and revised the manuscript. S.X., M.C., and P.G. were responsible for data collection and data analysis, and revised the manuscript. Z.D., T.F., and L.Z. contributed to data analysis and interpretation. W.T., L.Z., X.F., and S.L. participated in data analysis and manuscript revision. All authors have given final approval for the manuscript to be published and have agreed to be responsible for all aspects of the manuscript.

DECLARATION OF INTERESTS
The authors declare no competing interests.

SUPPLEMENTAL INFORMATION
Supplemental information can be found online at https://doi.org/10.1016/j.xinn.2021.100141.

LEAD CONTACT WEBSITE
https://yulab-smu.top.