nature protocols

https://doi.org/10.1038/s41596-024-01020-z

Protocol

Check for updates

# Using clusterProfiler to characterize multiomics data

**Shuangbin Xu[1,2,5], Erqiang Hu[1,5], Yantong Cai[1,3,5], Zijing Xie[1,5], Xiao Luo[1,5], Li Zhan[1], Wenli Tang[1], Qianwen Wang[1], Bingdong Liu[1,4], Rui Wang[1], Wenqin Xie[1], Tianzhi Wu[1], Liwei Xie[4] & Guangchuang Yu[1,2,3]** ✉

## Abstract

With the advent of multiomics, software capable of multidimensional enrichment analysis has become increasingly crucial for uncovering gene set variations in biological processes and disease pathways. This is essential for elucidating disease mechanisms and identifying potential therapeutic targets. clusterProfiler stands out for its comprehensive utilization of databases and advanced visualization features. Importantly, clusterProfiler supports various biological knowledge, including Gene Ontology and Kyoto Encyclopedia of Genes and Genomes, through performing over-representation and gene set enrichment analyses. A key feature is that clusterProfiler allows users to choose from various graphical outputs to visualize results, enhancing interpretability. This protocol describes innovative ways in which clusterProfiler has been used for integrating metabolomics and metagenomics analyses, identifying and characterizing transcription factors under stress conditions, and annotating cells in single-cell studies. In all cases, the computational steps can be completed within ~2 min. clusterProfiler is released through the Bioconductor project and can be accessed via https://bioconductor.org/packages/clusterProfiler/.

## Key points

- clusterProfiler is a software package for characterizing and interpreting omics data. Functional enrichment can be achieved using either over-representation or gene set enrichment analyses; it supports the use of a variety of databases, e.g., Gene Ontology and Kyoto Encyclopedia of Genes and Genomes.
- Three procedures show specific R commands for example applications asking different research questions and having different graphical outputs. Advice is provided on how to modify the procedures for other applications.

## Key references

Yu, G. et al. *OMICS* **16**, 284–287 (2012): https://doi.org/10.1089/omi.2011.0118

Wu, T. et al. *Innovation* **2**, 100141 (2021): https://doi.org/10.1016/j.xinn.2021.100141

Ne, M. et al. *Nat. Commun*. **12**, 6479 (2021): https://doi.org/10.1038/s41467-021-26685-y

Alexandre, P. A. et al. *Genome Biol.* **22**, 273 (2021): https://doi.org/10.1186/s13059-021-02489-7

Sankowski, R. et al. *Nat. Med*. **30**, 186–198 (2024): https://doi.org/10.1038/s41591-023-02673-1

[1]Department of Bioinformatics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, China. [2]Division of Laboratory Medicine, Microbiome Center, Zhujiang Hospital, Southern Medical University, Guangzhou, China. [3]Dermatology Hospital, Southern Medical University, Guangzhou, China. [4]State Key Laboratory of Applied Microbiology Southern China, Guangdong Provincial Key Laboratory of Microbial Culture Collection and Application, Guangdong Open Laboratory of Applied Microbiology, Guangdong Institute of Microbiology, Guangdong Academy of Sciences, Guangzhou, China. [5]These authors contributed equally: Shuangbin Xu, Erqiang Hu, Yantong Cai, Zijing Xie, Xiao Luo. ✉e-mail: gcyu1@smu.edu.cn

## Introduction

### Application of the protocol

Functional enrichment analysis is a broad approach used to investigate the association of specific gene lists or sets with certain biological functions, pathways or classifications[1]. This analysis is performed computationally and is commonly used to study important patterns observed in gene expression, protein expression or other large-scale bioinformatics data. The two most common approaches to functional enrichment analysis are:

(1) Over-representation analysis (ORA)[2], is a method primarily used on a predefined list of genes, such as significantly differentially expressed genes (DEGs) from an experiment. ORA determines which functionalities or pathways appear at a higher frequency than expected in the entire genome or a reference set, making it most suitable for analyzing genes with substantial effects.

(2) Gene set enrichment analysis (GSEA)[3] evaluates the entire gene expression profile, unlike ORA, to determine whether a gene set is prominently ranked. In this context, a gene set might be a collection of genes connected with a particular biochemical pathway or associated with specific physiological functions, disease processes, or pharmacological responses. GSEA identifies collective gene behavior even if individual genes show minor changes.

Despite the widespread utilization of both GSEA and ORA, many software platforms are predominantly focused on conventional pathway enrichment assessments[4]. These platforms often confine their analyses to mainstream scenarios, such as alterations in gene expression patterns in response to stimuli, mapping common signaling pathways and investigating the general molecular mechanisms associated with specific diseases. Unfortunately, this narrow focus tends to bypass potential discoveries in burgeoning domains within molecular biology and bioinformatics, demanding a more interdisciplinary and nuanced approach for exploration[5,6].

clusterProfiler is a versatile tool that seamlessly integrates both GSEA and ORA methodologies. Its ability to allow users to customize databases and annotations offers a broader interpretative range, facilitating investigations in emerging areas from single-cell types to bacterial metabolomes and even transcription factor analyses. In this protocol, we demonstrate the universal and flexible capabilities of clusterProfiler for performing comprehensive functional enrichment analysis. For this, the key areas of focus are:

- Unraveling the intricate interconnections between microbiota, metabolites and diseases
- Pinpointing the active transcription factors in the plant cold-resistance pathway
- Annotating cell types for single-cell transcriptomics

By broadening the spectrum of functional enrichment analysis applications, we aim to foster the development of innovative software and analytical strategies for future research horizons.

### Development of the protocol

ClusterProfiler was initially introduced in 2012 (ref. 7) with the primary function of performing ORA for a variety of model organisms, allowing functional profiling comparisons, such as different treatment groups. Over time, the software has undergone continuous updates and maintenance to enhance the user experience. Notably, the software[8] underwent four major upgrades to further enhance its capabilities:

1. The analysis methods were expanded to include both ORA and GSEA.
2. The annotation libraries supported by the software became richer and more diverse.
3. New auxiliary modules were integrated to enhance the depth of functional analysis.
4. The user interface for enrichment analysis was optimized to be more concise and intuitive.

Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) are two extensively used tools in biological research, each focusing on different biological domains[9]. GO provides detailed information about gene functions, covering the roles of gene products within cells, while KEGG focuses on unveiling the overall operation of metabolic pathways and signal transduction networks[10]. By integrating these two useful annotation libraries, the clusterProfiler package greatly expands its capabilities, supporting ORA and GSEA to offer extensive explanations and comprehensive annotations for biological questions in various research scenarios. This expansion enhances the flexibility and depth of clusterProfiler in addressing a wide range of application issues, making it an indispensable tool in the field of functional enrichment analysis.

In the current release of clusterProfiler, there has been a substantial improvement in data annotation sources. This version not only supports access to Bioconductor's OrgDb for the latest annotations of model organisms, updated semi-annually[11], but also introduces new data interfaces for nonmodel organisms, such as those available through platforms such as AnnotationHub[12]. Additionally, to meet the growing need for exploring functional characteristics from multiple perspectives and the demand for advanced molecular and cellular analysis, clusterProfiler has integrated interfaces for several other databases, including MSigDB[13] and WikiPathways[14]. To facilitate the integration of analyses from selected databases or to incorporate analysis results from tools like Blast2GO[15] and KAAS[16], the new version provides two interfaces, enricher and GSEA. These improvements make clusterProfiler a more comprehensive and powerful tool to support a wide range of bioinformatics analysis and research.

By integrating our developed packages such as ReactomePA[17], meshes[18], DOSE[19], GOSemSim[20] and ChIPseeker[21,22] into the framework of clusterProfiler, we have extended the software's capability to conduct detailed analyses across multiple dimensions, offering insights into biological processes, disease mechanisms and functional pathways. ReactomePA serves as a powerful tool for exploring and analyzing biological pathways within the Reactome database; the meshes package adds the functionality of enrichment analysis for literature and biomedical concepts, and the DOSE package is focused on disease ontology enrichment analysis, facilitating the exploration of connections between genes and diseases. Moreover, ChIPseeker broadens the application of clusterProfiler to include enrichment analysis of noncoding regions and epigenomic data, which is crucial for understanding gene expression regulation and epigenetic mechanisms. Additionally, GOSemSim optimizes result analysis by removing redundancies from GO enrichment results, simplifying the biological interpretation process, and enhancing the precision of the analysis.

The clusterProfiler also has enhanced data processing and visualization capabilities achieved by incorporating the tidyverse[23] framework and the enrichplot package[24]. The inclusion of tidyverse, with tools such as dplyr, offers a smoother approach to data manipulation, making the handling of datasets more efficient. Similarly, the enrichplot package, leveraging the power of ggplot2 (ref. 25), expands the variety of visualization formats and options available. For example, it introduces visualization methods such as enrichment maps, which effectively depict the relationships between enriched analysis pathways[26]. Furthermore, it enables tiered clustering of enrichment analysis results, providing users with a comprehensive overview of the outcomes.

### Advantages of clusterProfiler

ClusterProfiler enhances its utility by integrating additional functionalities, expanding annotation database interfaces, incorporating auxiliary packages and streamlining the data analysis and visualization processes. These features enable clusterProfiler to be widely applied to single-cell and other omics data, providing some interpretability for potential biological issues. For instance, R. Hoover and colleagues utilized the enrichGO function for ORA to investigate the role of localized ablative immunotherapy in controlling tumor growth, identifying regulatory DEGs related to localized ablative immunotherapy's molecular mechanisms and revealing selective stimulation of T cell activation and IFN-related pathway genes in CD8$^+$ T cell subpopulations[27]. In research exploring core genes associated with immune cell infiltration and gastric cancer prognosis[28], functions such as gseKEGG and gseGO were employed to delve into prognostic molecular markers and immune response mechanisms, uncovering significant enrichments in cancer response pathways such as NF-kB, lymphocyte and CD40 pathways.

Over the past 12 years, we have promptly addressed feedback and resolved issues through various channels, including Bioconductor forum and GitHub issues. To closely align with user needs, we have provided an exhaustive and detailed user manual, supplemented by a wealth of tutorial materials (https://yulab-smu.top/biomedical-knowledge-mining-book/). Researchers can select and customize unique analysis scenarios suited to their projects. For example, when analyzing a specific type of cancer, they can draw inspiration from the workflow of another cancer with similar biological traits[29].

## Comparative analysis of alternative tools

Although many software tools are available for enrichment analysis, they all have some limitations, such as Enrichr[30], WebGestalt[31] and fgsea[32], which do not support full-species analysis or the latest KEGG annotation updates. Tools such as GOstats[33], gprofiler2[34], DAVID[35], Enrichr[30] and Metascape[36] only support the ORA algorithm and cannot handle gene ranking lists without specific thresholds. Most tools do not support visualization or only offer limited visualization options, making it challenging to handle different usage scenarios and comprehensively analyze enrichment results. clusterProfiler overcomes these limitations and offers additional advantages, such as support for the tidy interface, gene ID conversion and user-customized annotation datasets in formats including GMT, gson and data frame. Moreover, clusterProfiler's results can be integrated with ggplot2 for more flexible and extensive visualization options. The comparative results are detailed in Supplementary Table 1.

Another often overlooked strength of clusterProfiler, compared with other enrichment tools, is its scalability. This is not just limited to the continuously expanding selection of its functional modules. So far, it has been integrated into over 40 renowned CRAN and Bioconductor packages (Table 1) and has found its place in various advanced computational pipelines and online platforms such as ViralLink[37] and ICARUS[38], highlighting its substantial utility and growing influence in the scientific community.

## Applications

clusterProfiler can be applied to the intricate analysis of many different omics datasets. Particularly in the field of cancer research, it aids researchers in gaining deeper insights into the relationships between gene mutations or differential expressions and cancer development[39]. Additionally, in evolutionary genomics, this tool unveils functional categories or pathways related to specific evolutionary events[40]. In the realm of metabolomics, it is applied for analyzing pathways associated with differential metabolites and for integrative analysis with transcriptomic data[41]. For microbiomics, it analyzes the functional characteristics of microbial communities[42]. In the burgeoning realm of single-cell transcriptomics, including its spatial dimension, clusterProfiler has proven to be an invaluable tool for data analysis and interpretation[43].

This research protocol aims to provide a comprehensive analysis of the diverse application scenarios of clusterProfiler (Fig. 1) using three specific examples:

1. Comparing functional profiles across diseases.
2. Transcription factor analyses in nonmodel organisms.
3. Automated cell type annotation from single-cell transcriptomic data.

### Comparing functional profiles across diseases

In disease research, combining multiomics data is key to deeply understanding biological systems' complexities and investigating disease causes. Metagenomics predominantly concentrates on the genetic constituents of microorganisms, whereas metabolomics centers on the metabolic end products of the host–microbe interactions. The confluence of these two fields affords a distinctive vantage point, enabling an all-encompassing examination of disease complexity and its nuanced subtypes. Within this framework, we employed clusterProfiler to conduct functional enrichment analyses on the differential metabolites and differential microbial genes of two subtypes of inflammatory bowel disease (IBD)[44] and then compared the results. This integrated analysis can provide valuable insights into the metabolic potential and functional capabilities of the microbial community under different disease conditions.

**Table 1 | R libraries that leverage clusterProfiler for their functional analysis tasks**

| Package | Description | repo |
|---|---|---|
| AutoPipe | Automated transcriptome classifier pipeline: comprehensive transcriptome analysis | CRAN |
| DRviaSPCN | Drug repurposing in cancer via a subpathway crosstalk network | CRAN |
| genekitr | Gene analysis toolkit | CRAN |
| Grouphmap | 'Grouphmap' is an automated one-step common analysis of batch expression profile | CRAN |
| immcp | Poly-pharmacology toolkit for traditional Chinese medicine research | CRAN |
| pathwayTMB | Pathway-based tumor mutational burden | CRAN |
| PMAPscore | Identify prognosis-related pathways altered by somatic mutation | CRAN |
| RVA | RNA sequencing visualization automation | CRAN |
| tinyarray | Expression data analysis and visualization | CRAN |
| TOmicsVis | Transcriptome visualization process scheme | CRAN |
| bioCancer | Interactive multiomics cancers data visualization and analysis | Bioconductor |
| CBNplot | Plot Bayesian network inferred from gene expression data based on enrichment analysis results | Bioconductor |
| CEMiTool | Co-expression modules identification tool | Bioconductor |
| CeTF | Coexpression for transcription factors using regulatory impact factors and partial correlation and information theory analysis | Bioconductor |
| debrowser | Interactive differential expresion analysis browser | Bioconductor |
| EasyCellType | Annotate cell types for single-cell RNA sequencing data | Bioconductor |
| eegc | Engineering evaluation by gene categorization (eegc) | Bioconductor |
| enrichTF | Transcription factors enrichment analysis | Bioconductor |
| esATAC | An easy-to-use systematic pipeline for transposase-accessible chromatin sequencing (ATACseq) data analysis | Bioconductor |
| famat | Functional analysis of metabolic and transcriptomic data | Bioconductor |
| fcoex | Fast correlation-based filter algorithm (FCBF)-based co-expression networks for single cells | Bioconductor |
| GDCRNATools | an R/Bioconductor package for integrative analysis of long non-coding RNA (lncRNA), mRNA and microRNA (miRNA) data in Genomic Data Commons (GDC) | Bioconductor |
| goSorensen | Statistical inference based on the Sorensen–Dice dissimilarity and the GO | Bioconductor |
| IRISFGM | Comprehensive Analysis of Gene Interactivity Networks Based on Single-Cell RNA-Seq | Bioconductor |
| MAGeCKFlute | Integrative analysis pipeline for pooled clustered regularly interspaced short palindromic repeats functional genetic screens | Bioconductor |
| MetaPhOR | Metabolic pathway analysis of RNA | Bioconductor |
| methylGSA | Gene set analysis using the outcome of differential methylation | Bioconductor |
| MicrobiomeProfiler | An R/shiny package for microbiome functional enrichment analysis | Bioconductor |
| miRspongeR | Identification and analysis of miRNA sponge regulation | Bioconductor |
| MoonlightR | Identify oncogenes and tumor suppressor genes from omics data | Bioconductor |
| multiSight | Multiomics classification, functional enrichment and network inference analysis | Bioconductor |
| PanomiR | Detection of miRNAs that regulate interacting groups of pathways | Bioconductor |
| PFP | Pathway fingerprint framework in R | Bioconductor |
| Pigengene | Infers biological signatures from gene expression data | Bioconductor |
| seqArchRplus | Downstream analyses of promoter sequence architectures and HTML report generation | Bioconductor |
| signatureSearch | Environment for gene expression searching combined with functional enrichment analysis | Bioconductor |
| TimiRGeN | Time sensitive microRNA–mRNA integration, analysis and network generation tool | Bioconductor |
| ExpHunterSuite | Package for the comprehensive analysis of transcriptomic data | Bioconductor |
| maEndToEnd | An end-to-end workflow for differential gene expression using Affymetrix microarrays | Bioconductor |
| recountWorkflow | recount workflow: accessing over 70,000 human RNA sequencing samples with Bioconductor | Bioconductor |
| TCGAWorkflow | TCGA workflow analyze cancer genomics and epigenomics data using Bioconductor packages | Bioconductor |

### Transcription factor analyses in nonmodel organisms

In light of recent advancements in sequencing methodologies, the scientific community has been endowed with enhanced capabilities to elucidate the intricacies of nonmodel organisms. Contemporary nonmodel species have now been cataloged with extensive gene

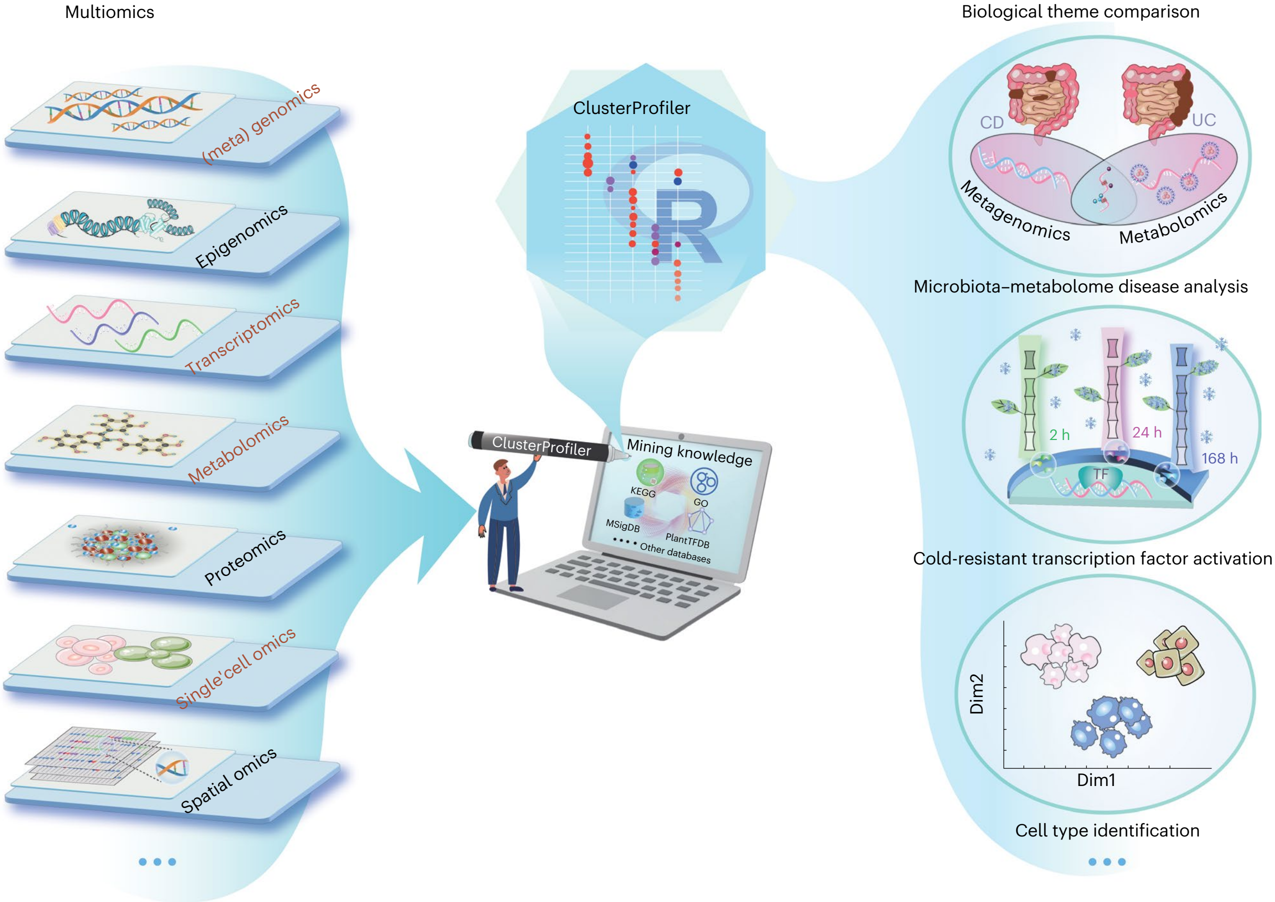


**Fig. 1 | Overview of the protocol.** As a functional enrichment analysis tool, clusterProfiler facilitates the systematic exploration of biological mechanisms and the characterization of biological states across a wide array of species. It accepts multiomics data, offering a clean and intuitive interface that enables researchers to efficiently access, manage and visualize enrichment analysis results. Uniquely, it supports the simultaneous analysis and comparison of data from various treatments and timepoints within a single computational operation. This feature streamlines the identification of functional consensus and the detection of discrepancies across diverse conditions. Detailed applications in this protocol include synchronized analysis of metabolomes and metagenomes, identification and functional characterization of transcription factors under stress conditions, and cell annotation within single-cell research paradigms. Dim, dimension.

annotation databases, thereby facilitating a more lucid comprehension of cellular system biology attributes, encompassing gene regulatory dynamics, cellular fate determinants and reaction paradigms to biological provocations. In this discourse, we delineate a methodological framework to harness the transcriptomic sequencing datasets derived from the nonmodel entity, bamboo[45]. This involves the strategic integration of a specialized plant transcription factor database for GSEA. Our investigative objective centers on pinpointing transcriptional regulators that are operational at disparate developmental junctures, further aiming to decode their functional implications in the context of cold acclimation responses.

### Automated cell type annotation from single-cell transcriptomic data

Single-cell sequencing methodologies have empowered researchers with the capability to scrutinize gene expression profiles at an individual cell granularity, thereby probing the intrinsic cellular heterogeneity with remarkable precision[46]. Such meticulous levels of data interrogation have ushered in a new era of depth in biological research. Nevertheless, in contemporary single-cell analytical paradigms, the comprehensive potential of the clusterProfiler tool appears to remain underutilized. Predominantly, its utility has been harnessed for functional elucidation of DEGs and delineation of associated biological pathways. In addressing cell type annotation, characterized by known genes or limited datasets yet requiring accuracy, the proclivity among researchers seems to gravitate toward manual annotation. This approach, albeit pragmatic, becomes intricate especially when annotating cells that belong to an overarching type but exhibit diverse subtypes, given the propensity of such subtypes to manifest overlapping gene expression signatures. It is fortuitous that the research community now has access to a plethora of meticulously annotated gene databases, such as the C8 gene set from MSigDB[47], for cell type annotation; these databases provide researchers with genes highly expressed in different cell types from previous studies, serving as a reference for manual annotation methods. Furthermore, annotation approaches for single-cell data now closely resemble those used in conventional differential gene expression analysis. Drawing from this insight, we leveraged the C8 gene set from MSigDB, synergistically integrating it with the clusterProfiler tool. This method helps to avoid artificial biases and can also serve as a foundation for manual annotations.

## Experimental design

As mentioned earlier, within these three examples, we primarily employ clusterProfiler's ORA (used in all three examples) and the GSEA algorithm (utilized in the second example) to explore downstream biological pathway regulation and cell type identification.

For ORA, the overall approach involves:

- Obtaining a list of features of interest, such as differential genes or metabolites, through differential analysis
- Selecting an appropriate annotation database, such as GO or the KEGG pathway database
- Using the enrichment functionalities provided by clusterProfiler, such as enrichGO, enrichKEGG or enricher, to perform analysis using the ORA algorithm

For GSEA, the objective is to acquire a sorted list of features that hold biological significance, such as those sorted by fold-change values, *T* statistics or corrected *P* values from a between-group analysis. Then, utilize functions such as gseKEGG, gseGO or GSEA from clusterProfiler for the GSEA analysis. For scenarios involving multiple groups, the compareCluster function can be used for comparative analysis. Box 1 provides an explanation on how to choose between these these two methods.

Users can test each example using the provided data or adapt the analysis process based on their data. Identify a sample dataset similar to users or an output that aligns with their research findings. Additionally, perform preliminary upstream analysis before inputting data for these analyses. Our suggestions under each section can guide users in adapting the analysis process to fit their needs.

### Procedure 1: functional enrichment analyses in metabolomic and metagenomic data

In the example, we use the preprocessed KEGG gene abundance information from the metagenome and the metabolite feature abundance table from the metabolome. How to derive these two omics feature abundance tables from raw sequencing data is not the focus of this protocol, but we have summarized a general processing flow in Box 2. In the KEGG gene abundance table, the row is named after the KEGG gene ID (KO), and the column is named after the sample. In the metabolite feature abundance table, the row name is KEGG COMPOUND and the column name is also the sample name. Additionally, prepare a sample grouping information table where the row name is the sample name, ensuring it aligns with the column name of the feature abundance table. If the rows and columns in your data are exactly reversed, you may need to use the t function in R to transpose the feature abundance table.

In Step 3 of this example, we use the mp_diff_analysis method provided by MicrobiotaProcess[42] for differential analysis, which combines the Wilcoxon rank sum test and the linear discriminant analysis method to identify differential features. You might also consider other differential analysis methods, such as parametric tests such as the *t*-test or variance test for metabolome data, or simply using the Wilcoxon rank sum test. For identifying differential features in metagenomic functional genes, you could directly use the Wilcoxon rank sum test or employ tools such as edgeR[48] or ALDEx2 (ref. 49). Next, we used enrichKEGG and compareCluster, specifying the organism parameters as 'ko' and 'cpd' respectively, to perform

BOX 1

## The choice of GSEA or ORA

This box explains the selection between GSEA and ORA analysis, guiding users to choose according to their input data and analytical objectives. When selecting between these two enrichment methods, we typically base our decision on the characteristics of the input data and the analytical objectives. For input data, the ORA algorithm needs just a list of genes of interest, which could include differential genes, single-cell marker genes or proteins from immunoprecipitation interactions. On the other hand, GSEA requires a full, ranked gene list, such as those ordered by log2 fold change. Regarding analysis goals, ORA employs Fisher's exact test to check if a set of genes is notably present in a gene set, a typical approach in enrichment analysis. GSEA assesses if a gene set tends to cluster at the ranked list's top or bottom, identifying if the gene set is activated or suppressed under specific conditions. The strength of GSEA lies in its ability to spot subtle yet consistent pathways without depending solely on differential genes.

enrichment analysis on the differential features of the metagenome and metabolome. If you use the feature ID from another common database, such as Enzyme Commission (EC) numbers[50], Human Metabolome Database (HMDB)[51] or the Small Molecule Pathway Database (SMPDB)[52], you can use an in-house developed package, MicrobiomeProfiler, which provides enrichKO, enrichHMDB and enrichSMPDB to replace enrichKEGG function.

**Procedure 2: transcription factor analysis pertaining to cold tolerance in PE**

We use data including the original gene abundance expression table for Moso bamboo *Phyllostachys edulis* (PE) under cold conditions and control, where the row name is the gene name and the column name is the sample name. The method for obtaining this gene expression table is detailed in Box 3. Additionally, there is a sample information table, similar to the above, where the sample names need to match the sample names in the abundance table. To explore the potential gene transcriptional regulatory mechanisms of Moso bamboo (PE) under cold conditions, we downloaded a list of transcription factor families and a table of gene GO annotations for Moso bamboo from PlantTFDB[53]. Users can replace this transcription factor list and functional annotation table according to their own data and experimental design.

To identity the significantly enriched transcription factors, we calculated the total gene log fold-change values between different cold treatment durations and the control group and sorted these values as input for GSEA analysis. As mentioned above, users can consider other statistics, such as sorting genes based on *P* values. In addition, the log fold-change values of all genes are calculated using DESeq2 (ref. 54), without undergoing *P* value filtering

BOX 2

## Preprocessing for metabolomics and metagenome data

This box mainly describes the processing workflow for upstream data of metagenomics and metabolomics. Users can refer to it and, of course, they can also consult other protocols published for the upstream data processing of these two omics.

In the metagenomics section, initial quality control of raw reads was conducted using fastp[73], followed by the removal of host (human) DNA by aligning with the human genome with Bowtie2 (ref. 74), using MetaPhlAn2 for taxonomic classification[75], retaining species with relative abundances over 0.1%. Subsequently, functional analysis was performed using HUMAnN2 (ref. 76), organizing gene abundances by KEGG KO numbers or EC numbers.

In the metabolomics section, in general, the sample metabolites are analyzed using liquid chromatography–tandem mass spectrometry techniques, encompassing polar metabolites, lipids, free fatty acids and bile acids. Each technique employed high-sensitivity mass spectrometers capable of detecting both known and novel metabolites. Raw liquid chromatography–tandem mass spectrometry data processing, including noise reduction, peak detection, isotopic cluster detection, retention time alignment and metabolite identification through database searches can be performed using Genedata Expressionist software (https://www.genedata.com/products-services/expressionist).

## BOX 3

# Preprocessing for transcriptome data

This box outlines the key steps and tools used for preprocessing raw transcriptome sequence data.

Transcriptome analysis is important in modern biological and medical research, used to uncover gene regulatory networks and mechanisms responding to environmental changes or disease-related alterations. It is needed to convert the raw reads into gene expression information after raw sequence reads are obtained from sequencing platforms. This analysis process typically includes the following steps. First, quality control of sequencing reads, includes assessing the quality of sequencing reads, removing low-quality reads, eliminating adapters and filtering out contaminants. The software typically used for this step includes FastQC, Trimmomatic[77], or fastp[73]. Next, alignment and quantification, map the cleaned reads to a reference genome or transcriptome using tools such as Tophat[78], STAR[79], Bowtie2[74] or HISAT2 (ref. 80), etc, then quantify the transcripts with tools such as cuffquant[81], RSEM[82], featureCounts[83] or HTSeq[84], etc. based on the alignment results. After these steps, users obtain a gene expression count table, which also is analyzed in this protocol's second procedure.

in this example. Users can also opt to calculate using edgeR[48], limma[55] or other tools. Next, we utilized the enricher function from clusterProfiler for GO enrichment analysis on the target genes of transcription factors significantly enriched by GSEA, pinpointing the key biological processes of target genes regulated by these transcription factors. In this step, users can choose the appropriate knowledge database according to the biological questions they want to explore, such as the KEGG pathway or the Wikipathway database, as detailed in Box 4.

## BOX 4

# Frequently used annotation databases and their interface with clusterProfiler

This box outlines the features of popular knowledge databases, their potential applications and summarizes how the clusterProfiler package facilitates access to this data.

Popular databases for functional enrichment analysis are GO[85], KEGG[50], disease ontology (DO)[86], DisGeNET[87], Network of Cancer Genes (NCG)[88], MSigDb[13], WikiPathways[14] and Reactome pathway[89], among others. Each offers unique features and provides annotations for various biological perspectives.

- The GO database provides detailed annotations on gene functions, cellular components and biological processes through a hierarchical system, enabling in-depth functional categorization. This makes it ideal for analyzing gene functions and localizations within biological processes, particularly for initial comprehensive studies on gene functions
- KEGG offers detailed annotations on biochemical pathways and functions, including metabolic and signaling pathways, making it suitable for exploring gene roles within these pathways
- DO, DisGeNET and NCG provide annotations on the links between genes and diseases, including a hierarchical disease structure, ideal for examining the connections between genes and particular diseases. Notably, NCG focuses on gene-cancer associations, making it useful for researching gene roles in cancer development and progression
- WikiPathways delivers manually curated information on biological pathways, making it ideal for studying gene functions in particular pathways or fields
- Reactome Pathway offers comprehensive details on various biological processes, such as metabolism, signal transduction, gene regulation, and immune response. This is suitable for exploring gene functions within cellular processes, metabolic pathways and signal transduction, as well as gaining insights into gene roles in specific biological processes.

Numerous annotation libraries are accessible, and their selection should align with specific needs and the biological inquiries being addressed. However, users must consider the release dates of these databases, since using outdated versions can adversely affect the outcomes.

clusterProfiler and its suite of packages provide functions to interface with multiple annotation databases, such as the enrichGO and gseGO functions of clusterProfiler for interfacing with the GO database, enrichKEGG and gseKEGG for interfacing with the KEGG database and the enrichDO and gseDO functions of the DOSE package for interfacing with the DO database. For other annotation databases, clusterProfiler offers generic enrichment analysis functions and can accommodate various input formats, including GMT, gson (https://cran.r-project.org/package=gson) and data frames of gene-to-gene set mappings.

### Procedure 3: single-cell transcriptomic cell type annotation

In this example, we used single-cell data from peripheral blood mononuclear cells (PBMCs), generated by Cellranger[56], along with the C8 cell type signature gene sets from the MSigDB database. We primarily used Seurat for upstream data reading and preliminary analysis, including the removal of mitochondrial genes, data normalization, identification of highly variable genes, principal component analysis (PCA), uniform manifold approximation and projection (UMAP) dimensionality reduction and clustering analysis. These steps mainly prepare for subsequent cell dimensionality reduction visualization and marker gene identification for cell clusters. For these upstream analyses, users can also consider using scater, scuttle and scran from Bioconductor[57,58] or scanpy in Python[59]. For detailed steps, please refer to the official documentation of these software tools, which we will not describe in detail here.

Next, we used the RunMCA analysis and GetGroupGeneSet to extract the top 20 characteristic genes from each cell cluster by CelliD[60]. We then used clusterProfiler's compareCluster and enricher (a general ORA function provided by clusterProfiler) to perform cell GSEA for each cluster. Finally, we annotated each cell cluster with the most significantly enriched cell type based on the cell type signature gene sets organized in the MSigDB database. Users can also use their reliable cell type gene set lists collected independently.

By following these guidelines and making necessary adjustments based on the characteristics of user data and research objectives, users can maximize the utility of clusterProfiler in their studies.

## Limitations

### Selecting the appropriate annotation database

In the process of conducting functional enrichment analysis using clusterProfiler, the selection of an appropriate annotation database becomes a crucial step. Users must recognize that various databases might exhibit substantial disparities in version update frequencies and the gene sets they encompass. Opting for inaccurate or outdated annotation information can introduce profound and irreversible biases to the conclusions of research analysis[61,62]. Additionally, given that clusterProfiler largely relies on public annotation resources, its functionality and accuracy are substantially constrained by the choice of database. Consequently, when determining which annotation database to employ, researchers should thoroughly evaluate the applicability of its data, the timeliness of updates and its alignment with the research topic to ensure the precision and reliability of the analytical output. More detail on the introduction to commonly used databases can be found in Box 4.

### Depth and breadth of annotations

In the process of gene enrichment analysis, the 'depth' and 'breadth' of annotations often serve as decisive factors, yet they inherently bear certain limitations[1]. First, it must be acknowledged that the current depth of gene–function associations might be superficial. Some studies indicate that numerous gene–function relationships are grounded in preliminary experimental evidence, lacking a deeper biological context or detailed mechanistic research. This could be attributed to the fact that many gene–function associations are based on initial data from high-throughput experiments, which, although capable of providing a wealth of information, might contain false positives or not be comprehensive enough.

Second, the 'breadth' of annotations brings along a series of issues, particularly when considering the inconsistencies between different databases. This inconsistency might stem from different databases adopting various annotation standards and methods, as well as disparities in data update timings and frequencies. For instance, a gene might be annotated as being involved in a certain biological pathway in one database but lacks corresponding annotations in another. This not only potentially leads to variability in results, but also adds to the complexity of interpreting the results.

Therefore, when utilizing clusterProfiler or other gene enrichment analysis tools, researchers need to thoughtfully select and use annotation databases, while also being aware of the limitations of these tools and data. Wherever possible, researchers should consider validating and comparing data and annotations from multiple sources to attain more comprehensive and accurate results. The current mainstream gene set file format, GMT, only contains the correspondence between genes and gene sets. A large amount of meta data, such as the source and release time of the data, cannot be stored in this file format. This maybe one of the reasons why a large number of outdated annotation files are spread and used. To address this issue, we propose the gson file format (https://cran.r-project.org/package=gson) for storing gene sets and their related meta data. In the future, clusterProfiler will also be developed based on this format.

## Materials

### Software

- R 4.3.1 (https://cran.r-project.org/)
- Bioconductor 3.18 (https://www.bioconductor.org/install/)
- aplot 0.2.2 (https://cran.r-project.org/package=aplot)
- ggfun 0.1.4 (https://cran.r-project.org/package=ggfun)
- ggplot2 3.4.4 (https://cran.r-project.org/package=ggplot2)
- Seurat 5.0.1 (https://cran.r-project.org/package=seurat/)
- tidyr 1.3.1 (https://cran.r-project.org/package=tidyr)
- yulab.utils 0.1.4 (https://cran.r-project.org/package=yulab.utils)
- CelliD 1.10.1 (https://bioconductor.org/packages/CelliD/)
- clusterProfiler 4.10.0 (https://bioconductor.org/packages/clusterProfiler)
- DESeq2 1.42.0 (https://www.bioconductor.org/packages/DESeq2)
- enrichplot 1.22.0 (https://www.bioconductor.org/packages/enrichplot)
- ggsc 1.0.2 (https://bioconductor.org/packages/ggsc/)
- MicrobiotaProcess: 1.14.0 (https://bioconductor.org/packages/MicrobiotaProcess/)

### Equipment setup

#### Installing software

The R statistical computing environment is required to run the protocol. Users need to download and install R from https://cran.r-project.org/. To install the required packages, start an R session and run the following commands:

```
pkgs <- c("aplot", "CelliD", "clusterProfiler", "DESeq2",
 "enrichplot", "ggfun", "ggplot2", "ggrepel",
 "ggsc", "MicrobiotaProcess", "Seurat")
install.packages("BiocManager")
BiocManager::install(pkgs)
```

For macOS users, please make sure to have Xcode and Fortran compiler installed as they are necessary for some dependencies.

#### Data

The example data for running the protocol are compiled on https://yulab-smu.top/clusterProfiler_protocol and available for download (see also the 'Data availability' section). Detailed descriptions and advice for downloading these data are listed under separate headings below.

#### Functional enrichment analyses in metabolomic and metagenomic data

Paired metabolomic and metagenomic data were sourced from the supplementary files of a prior IBD study[44]. We have outlined the upstream data processing for metagenomes and metatranscriptomes in Box 2. It included 56 control samples, 76 ulcerative colitis (UC) patients and 88 Crohn's disease (CD) patients. Based on this dataset, we conducted functional enrichment analyses, aiming to identify shared and unique biological pathways and functional subsets that are perturbed in the two subtypes of IBD.

1. Download the processed metabolite and microbial gene abundance matrix and the experimental group information by typing the following into R:

```
url <- "https://yulab-smu.top/clusterProfiler_protocol/examples"
IBD_files <- c("mg.meta.csv", "mg.expr.csv",
 "metabolism_meta.csv", "metabolism_expr.csv")
for (f in IBD_files) {
 download.file(file.path(url, "IBD_2_subtypes", f), destfile = f)
}
```

### Functional enrichment analysis of cold-resistant transcription factors in PE

The transcriptomic dataset was derived from Moso bamboo (PE), which was meticulously cultivated under cold-stress conditions maintained at 4 °C. Sequential leaf samplings for in-depth transcriptomic sequencing were conducted at precise timepoints: 0, 2, 24 and 168 h postexposure. The pertinent data can be accessed through the dedicated repository: https://db.cngb.org/search/project/CNP0002243/. Furthermore, for intricate annotations associated with the regulatory dynamics of plant transcription factors, our reference point was the esteemed plantRegMap database[63]. Detailed insights into this can be garnered from: http://planttfdb.gao-lab.org/.

1. Download the PE transcriptome gene expression matrix (raw count) and the experimental group information by typing the following into R:

```
count_files <- c("counts.txt", "group_info.txt")

url <- "https://yulab-smu.top/clusterProfiler_protocol/examples/Phyllostachys_heterocycla"

for (f in count_files) {
 download.file(file.path(url, f), destfile = f)
}
```

2. Download PE annotation data including transcription factor regulatory relationships, transcription factor family memberships and GO annotations:

```
annot_files <- c("regulation_from_motif_CE_Phe.txt",
 "Phe_TF_list.txt", "Phe_GO_annotation.txt"
)
for (f in annot_files) {
 download.file(file.path(url, "annot_data", f), destfile = f)
}
```

### Single-cell transcriptomic data and its annotation for cell type identification

The single-cell dataset is sourced from the Seurat example, comprising single-cell sequencing of 2,700 PBMCs using the Illumina NextSeq-500 sequencing platform at: https://cf.10xgenomics.com/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz. The cell type annotation data is derived from MSigDB: https://www.gsea-msigdb.org/gsea/msigdb/human/collections.jsp#C8.

3. Download the single-cell dataset from the Seurat example by typing the following into R:

```
url <- "https://yulab-smu.top/clusterProfiler_protocol/examples/single_cell/"
dir.create("hg19")
pbmc_files <- c("barcodes.tsv", "genes.tsv", "matrix.mtx")
for (f in pbmc_files) {
 download.file(
 file.path(url, "filtered_gene_bc_matrices/hg19/", f),
 method = "auto",
 destfile = file.path("hg19", f)
)
}
```

4. Download gene-cell type annotation data from MsigDb:

```
download.file(
 file.path(url, "cell_marker_db/c8.all.v2023.1.Hs.symbols.gmt"),
 method = "auto",
 destfile = "c8.all.v2023.1.Hs.symbols.gmt"
)
```

## Procedure 1: metabolomics and metagenomics functional enrichment analysis

▲**CRITICAL** It is important to notice that each step outlined below includes a brief explanation followed by R codes. The provided codes are ready to be directly executed within the R command line interface.

▲**CRITICAL** Note that the 'Timing' refers to computational times in executing the R codes. The computational time for Procedure 1 as a whole is ~29.88 s.

▲**CRITICAL** Refer to Supplementary Note 2 for a version of the procedure where the syntax of the source code for the procedures is highlighted.

### Set up the environment and data objects

● **TIMING** ~12.42 s

1. Load the R packages into the R environment:

```
library(MicrobiotaProcess)
library(clusterProfiler)
library(ggplot2)
library(enrichplot)
```

2. Import metagenomic data:

```
meta_mg <- read.csv("mg.meta.csv")
metagenome <- read.csv("mg.expr.csv",
 row.names = 1,
 check.name = FALSE)
```

The first command reads the sample grouping information and the second command reads the gene abundance matrix.

◆ **TROUBLESHOOTING**

### Metagenomic data differential analysis

● **TIMING** ~41.16s

3. Define a function to perform differential analysis:

```
DA <- function(expr,
 meta,
 abundance = 'Abundance',
 group_colname = 'Diagnosis',
 force = TRUE,
 relative = FALSE,
 subset_group,
 diff_group,
 filter.p = ’pvalue’, …) {
 sign_group_colname <- paste0(’Sign_’, group_colname)
 mpse <- MPSE(expr)
 mpse <- mpse |> left_join(meta, by = "Sample")
 mpse |>
 dplyr::filter(!!as.symbol(group_colname) %in% subset_group) |>
 mp_diff_analysis(
 .abundance = !!as.symbol(abundance),
 .group = !!as.symbol(group_colname),
 force = force,
 relative = relative,
 filter.p = filter.p,
 …
) |>
 mp_extract_feature() |>
 dplyr::filter(!!as.symbol(sign_group_colname) == diff_group) |>
 dplyr::pull(OTU) |> suppressMessages()
}
```

This function uses an in-house developed R package, MicrobiotaProcess, to construct a Microbiota Processed Summarized Experiment (MPSE) object, which stores the abundance data of features and the sample metadata, then performs differential analysis and subsequently extracts significant results (based on *P* values). Details about certain parameters used during this procedure are provided in the supplementary material (for more details, see Supplementary Note 1).

4. Identify differentially abundant microbial genes:

```
groups <- c(CD = ’CD’, UC = ’UC’)
de_gene <- lapply(groups, function(x) {
 DA(expr = metagenome,
 meta = meta_mg,
 subset_group = c(x, ’Control’),
 diff_group = x)
})
```

Utilizing differential analysis methods to detect potential biomarkers in microbial communities associated with diseases has become a widely adopted strategy. Here, the first command creates a vector to store two types of samples (that is, subtypes of IBD). The second command uses lapply to separately analyze these two sample types versus the control group to identify enriched differential genes.

### Functional analysis of differential microbial genes

● **TIMING ~22.47 s**

5. Perform enrichment analysis of differential genes:

```
gene_enrich_result <- compareCluster(geneClusters = de_gene,
 fun = "enrichKEGG",
 organism = "ko")
```

This example uses the compareCluster function to carry out enrichment analysis on two IBD subtypes using the KEGG Orthology (KO) database. The KO database describes molecular

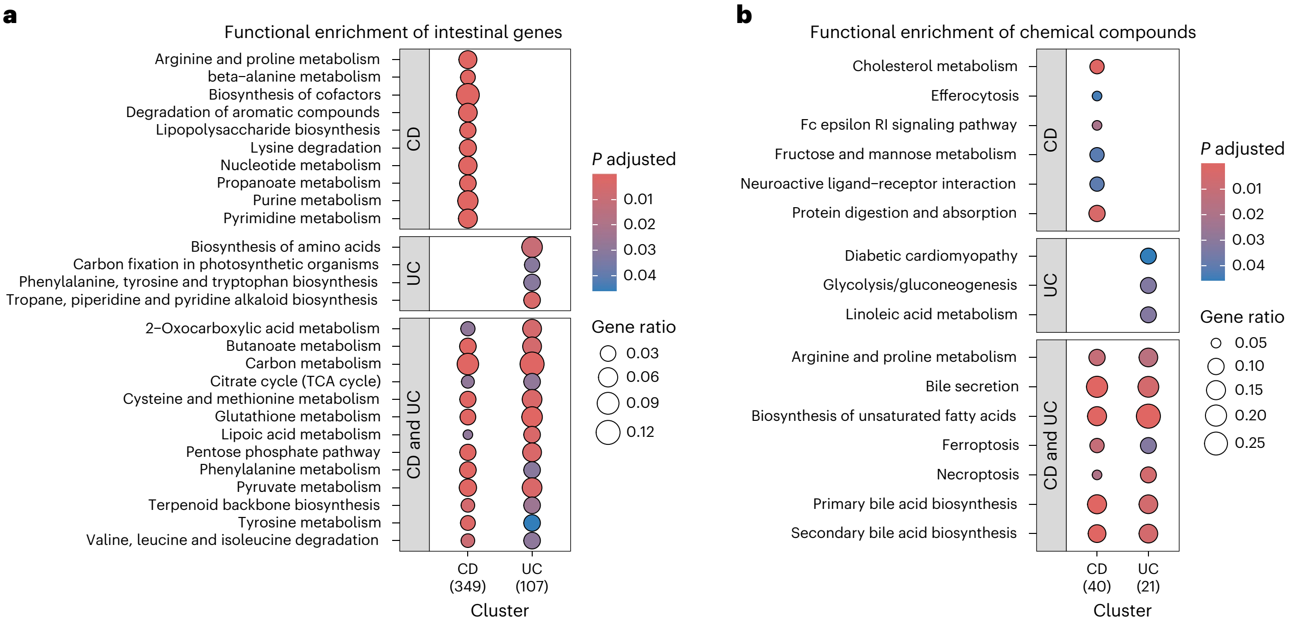


**Fig. 2 | Comparing functional profiles among distinct subtypes of IBD.** **a**,**b**, This study focuses on the intestinal microbiome genes (**a**) and intestinal metabolites (**b**). Enrichment analyses were conducted simultaneously for enriched genes and metabolites within both IBD subtypes using the compareCluster function and the results were subsequently visualized by the dotplot function. The *x* axis represents two IBD subtypes, while facets are used to depict shared or unique pathway enrichments.

functions in terms of functional orthologs. The output of this step is a comparative list of enriched pathways across the studied IBD subtypes, providing insights into shared and unique aspects of their pathophysiology. This step could be adapted if users want to use a different function with a different database.

6. Visualize the functional enrichment result of differential genes (Fig. 2a):

```
dotplot(gene_enrich_result, facet = ’intersect’, showCategory = 10,
 split = "intersect", label_format = 60) +
 ggtitle("Functional enrichment of intestinal genes") +
 theme(plot.title = element_text(hjust = 1))
```

In this example, the dotplot function is used to visualize the result with the ‘facet’ parameter set to ‘intersect’ to allow the separation of common and unique biological pathways that are enriched in different IBD subtypes. Other options for the ‘facet’ parameter include specifying a variable within a result, such as in KEGG analysis where the category variable indicates the categories of pathways, such as Cellular Processes, Organismal Systems and Human Diseases, etc. The ‘label_format’ parameter allows users to specify the display length of pathway identifiers, with automatic line wrapping if the length exceeds the specified limit.

## Metabolomic data differential analysis

● **TIMING** ~5.75 s

7. Import metabolomic data

```
meta_mb <- read.csv("metabolism_meta.csv")
metabolism <- read.csv("metabolism_expr.csv",
 row.names = 1,
 check.name = FALSE)
```

The first command reads the sample information and the second command reads the metabolite abundance information.

8. Identify differentially abundance metabolites

```
groups <- c(CD = 'CD', UC = 'UC')
de_cpd <- lapply(groups, function(x) {
 DA(expr = metabolism,
 meta = meta_mb,
 subset_group = c(x, 'Control'),
 diff_group = x)
})
```

Adhering to the same strategy, we utilize the DA function defined in the previous step to conduct differential analysis to identify significantly enriched differential metabolites in the two subtypes.

### Functional analysis of differential metabolites

● TIMING ~7.43 s

9. Perform enrichment analysis of differential metabolites

```
cpd_enrich_result <- compareCluster(geneClusters = de_cpd,
 fun = "enrichKEGG",
 organism = "cpd")
```

To conduct an enrichment analysis using the KEGG Compound database (CPD), which aids in characterizing biological processes relevant to the differential metabolites identified in the two IBD subtypes, follow the steps outlined below. While this protocol focuses on the CPD, users may opt to analyze their data with alternative databases such as HMDB or SMPDB for different perspectives on metabolite functions and pathways using the MicrobiomeProfiler package (see the 'Experimental design' section).

10. Visualize the functional enrichment results of differential metabolites (Fig. 2b):

```
dotplot(cpd_enrich_result, facet = 'intersect', showCategory = 10,
 split = "intersect", label_format = 60) +
 ggtitle("Functional enrichment of chemical compounds") +
 theme(plot.title = element_text(hjust = 1))
```

## Procedure 2: transcription factor analysis pertaining to cold tolerance in PE

● TIMING ~83.64 s

▲ CRITICAL The primary purpose of this procedure is to delineate the pipeline for analyzing next-generation sequencing datasets, rather than detailing the steps for processing raw sequencing data. Consequently, while we will not delve extensively into the methods used to convert raw data into analyzable datasets in this section, we will outline the minimum requirements needed to ensure that a dataset is suitable for the procedures that follow. Readers interested in this specific aspect are encouraged to consult the referenced literature[64]. We also have summarized the upstream analysis of the transcriptome in Box 3.

▲ CRITICAL The computational time for Procedure 2 as a whole is ~83.64 s.

▲ CRITICAL Refer to Supplementary Note 2 for a version of the procedure where the syntax of the source code for the procedures is highlighted.

### Prepare the expression matrix of Moso bamboo sequencing

● TIMING ~0.93 s

1. Load the R packages into the R environment:

```
library(DESeq2)
library(clusterProfiler)
library(ggplot2)
library(enrichplot)
library(aplot)
library(ggfun)
```

2. Read the expression matrix and sample grouping information:

```
counts <- read.delim("counts.txt") group_info <- read.delim("group_
info.txt") group_info$group <- factor(group_info$group, levels =
c("0h", "2h", "24h", "168h"))
```

The first command reads the expression matrix without normalization and the second command reads the experimental grouping data, in which the first column contains sample IDs, and the second column holds the sample grouping information, documenting the timepoint group information for each sample. Then we encode the group vector as a factor type by the factor function. Users should adjust the levels according to the order of their group information.

### Calculate logarithmic fold changes (log2FC) using DESeq2

● TIMING ~24.52 s

3. Create a DESeqDataSet object:

```
count_dds <- DESeqDataSetFromMatrix(countData = counts,
 colData = group_info,
 design = ~ group)
```

This command creates a DESeqDataSet object from the expression matrix and the sample grouping information which specifies the experiment design.

4. Use DESeq to perform default differential expression analysis, including logarithmic fold changes that incorporate data-driven prior distributions:

```
count_dds <- DESeq(count_dds)
```

This step performs normalization to mainly eliminate the differences between two groups of data caused by different library sizes and uses a negative binomial distribution to fit the data. Based on the expression level of genes and observed variance, it estimates the dispersion level and calculates the log2FC and *P* value for each gene.

### Extract log2FC, and sort genes based on the log2FC values

● TIMING ~1.95 s

5. Define groups (different timepoints) to be compared with the baseline group (i.e., 0 h):

```
time_points <- setNames(object = c("168h", "24h", "2h"),
 nm = c("168h_vs_0h", "24h_vs_0h", "2h_vs_0h"))
```

A grouping vector named 'time_points' is defined, where samples are categorized based on different timepoints (0, 2, 24 and 168 h). In this setup, each timepoint serves as a treatment group, with the 0 h group acting as a control group, against which other treatment groups (2, 24, 168 h) are compared.

6. Extract log2FC values:

```
all_result <- lapply(time_points, function(time_point) {
 result <- results(count_dds, tidy = TRUE,
 contrast = c("group", time_point, "0h"))
 setNames(object = result$log2FoldChange, nm = result$row) |>
 sort(decreasing = TRUE)
})
```

The results function extracts a result table from a DESeq analysis with the contrast parameter specifying which comparison to extract. The lapply function is utilized to iterate each timepoints to aggregate all the results in a named list. Since the GSEA algorithm requires a ranking of gene expression data associated with phenotypes, the log2FC values were extracted from the output table and sorted in decreasing order.

### Transcription factor enrichment analysis to identify perturbed transcription factors at different timepoints

● **TIMING ~27.33 s**

7. Prepare transcription factor annotation data:

```
tf_db <- read.delim(
 "regulation_from_motif_CE_Phe.txt", header = FALSE,
 colClasses = c("character", "NULL", "character", rep("NULL", 4))
) |> setNames(c("TF", "targetGene"))
```

The read.delim function reads the transcription factor annotation file by importing the first and third columns that contain the transcription factor gene IDs and the corresponding regulated target gene IDs, respectively. The setNames function sets the column names of the imported data.

8. Use compareCluster to perform batch GSEA for identifying perturbed transcription factors from each timepoint:

```
perturbed_TF_result <- compareCluster(all_result, fun = "GSEA",
 pvalueCutoff = .05,
 TERM2GENE = tf_db,
 seed = 1234)
```

The compareCluster function is utilized to perform GSEA on the sorted log2FC of each timepoint compared with the baseline group. In this process, the previously prepared gene annotation data, specifically the 'tf_db' data frame, is used. This data frame is passed to the TERM2GENE parameter of the compareCluster function, facilitating batch GSEA enrichment analysis during each comparison between timepoints and the baseline group. This is to identify the key transcription factors at different timepoints by evaluating target gene sets in the variations of gene expression. Unearth the connection between the regulatory functions of transcription factors and time. This step assists in delving deeper into the molecular mechanisms of transcription factors at different timepoints under cold conditions in PE.

9. Visualize the enrichment result using dotplot:

```
perturbed_TF_plot <- dotplot(perturbed_TF_result,
 showCategory = 25) +
 aes(shape = I(22)) +
 coord_flip() +
 theme_minimal() +
 theme_noxaxis() +
 xlab(NULL) +
 set_enrichplot_color(
c("#6C8FAD", "#84ADA7", "#C7B398"),
 .fun = ggplot2::scale_fill_gradientn
)
```

The result was visualized as a dot plot and the appearance of the plot was modified using the ggplot2 syntax including changing the shape and color scheme.
◆ TROUBLESHOOTING

### Predict the biological functions possibly regulated by the perturbed transcription factors

● TIMING ~27.48 s

▲ CRITICAL To further investigate the underlying biological mechanisms of PE response to cold, we perform GO enrichment analysis on the transcription factor-regulated genes using compareCluster.

10. Construct a named list of transcription factor (TF) target genes and extract a subset of the most significant TFs that are related to cold responses:

```
tf_id <- unique(get_plot_data(perturbed_TF_plot, "ID")[, 1])
tf_genes <- split(tf_db$targetGene, tf_db$TF)[tf_id]
```

The first command extracts the most significant transcription factors from the dot plot generated in the previous step. The second command constructs a named list from the TF target gene data frame and extracts the subset based on the output of the first command.

11. Import the GO annotations of PE genes:

```
go_db <- read.delim(file = "Phe_GO_annotation.txt")
```

The go_db object contains gene ID (first column) and corresponding GO annotation such as GO ID (second column), GO term (third column) and evidence code etc.

12. Characterize TF functions based on GO enrichment analysis of TF target genes:

```
TF_GO_result <- compareCluster(tf_genes, fun = "enricher",
 TERM2GENE = go_db[, c(2, 1)],
 TERM2NAME = go_db[, c(2, 3)])
```

The compareCluster function performs GO enrichment analysis on target genes regulated by different transcription factors.

13. Visualize TF function enrichment result:

```
TF_GO_plot <- dotplot(TF_GO_result, by = "count",
 showCategory = 3,
 label_format = 40) +
 theme_minimal() +
 theme(axis.text.x = element_text(vjust = 1, hjust = 1,
 angle = 30, size = 10)) +
 xlab(NULL) + ggtitle(NULL)
```

The dotplot function is used to visualize the enrichment results, with slight modification of the plot using the ggplot2 syntax.

### Visualization of transcription factor family annotation information

● **TIMING ~0.04 s**

14. Import TF family annotation information:

```
tf_family <- read.delim("Phe_TF_list.txt", row.names = 1)
```

The output object contains two columns including transcription factor gene ID and corresponding family information.

15. Prepare TF family data for visualization:

```
family_data <- subset(tf_family, Gene_ID %in% tf_id)
family_data <- family_data[order(family_data$Family),]
family_data$Gene_ID <- factor(family_data$Gene_ID,
 levels = family_data$Gene_ID)
```

The first command extracts a subset of TF family information based on the most significant TFs. The second command sorts the data based on the TF family names and the third command fixes the order for visualization.

16. Visualize the TF family information:

```
tf_family_plot <- ggplot(data = family_data,
 aes(x = Gene_ID, y = 1, fill = Family)) +
 geom_tile() +
 scale_fill_discrete(type=c('#B3D3AA', '#4B6B5C', '#E88A71',
 '#DEAB76', '#CD574D', '#85C1BF', '#BF38AE',
 '#176D84', '#7D83B7', '#4040C6', '#994B41')) +
 ggfun::theme_nothing()
```

The ggplot2 commands are utilized to visualize the family information with customized colors and all the theme elements were removed by ggfun.

### All-in-one integration to reveal transcription factor perturbation and subsequent biological effects

● **TIMING ~1.33 s**

17. Create a composite plot that integrates all the pieces of information (Fig. 3):

```
insert_top(tf_family_plot, perturbed_TF_plot, height = 5) |>
 insert_bottom(TF_GO_plot, height = 50)
```

The insert_top function is used to place the transcription factor enrichment dot plot at the top of the transcription factor family annotation block diagram. The assembly result from the insert_top function is then passed to the insert_bottom function through a pipe operator, positioning the GO enrichment dot plot of target genes regulated by the transcription factors at the bottom. The functions used to create this composite figure are available in the in-house developed package, aplot. It is more powerful than other tools because it can reconcile axes, not just assemble images. The perturbed TF, family information and target gene functions are perfectly aligned in the final figure and aid in the interpretation of biological mechanisms. In this instance, we apply both GSEA and ORA analysis. Please refer to Box 1 for an explanation of choosing these two methods.

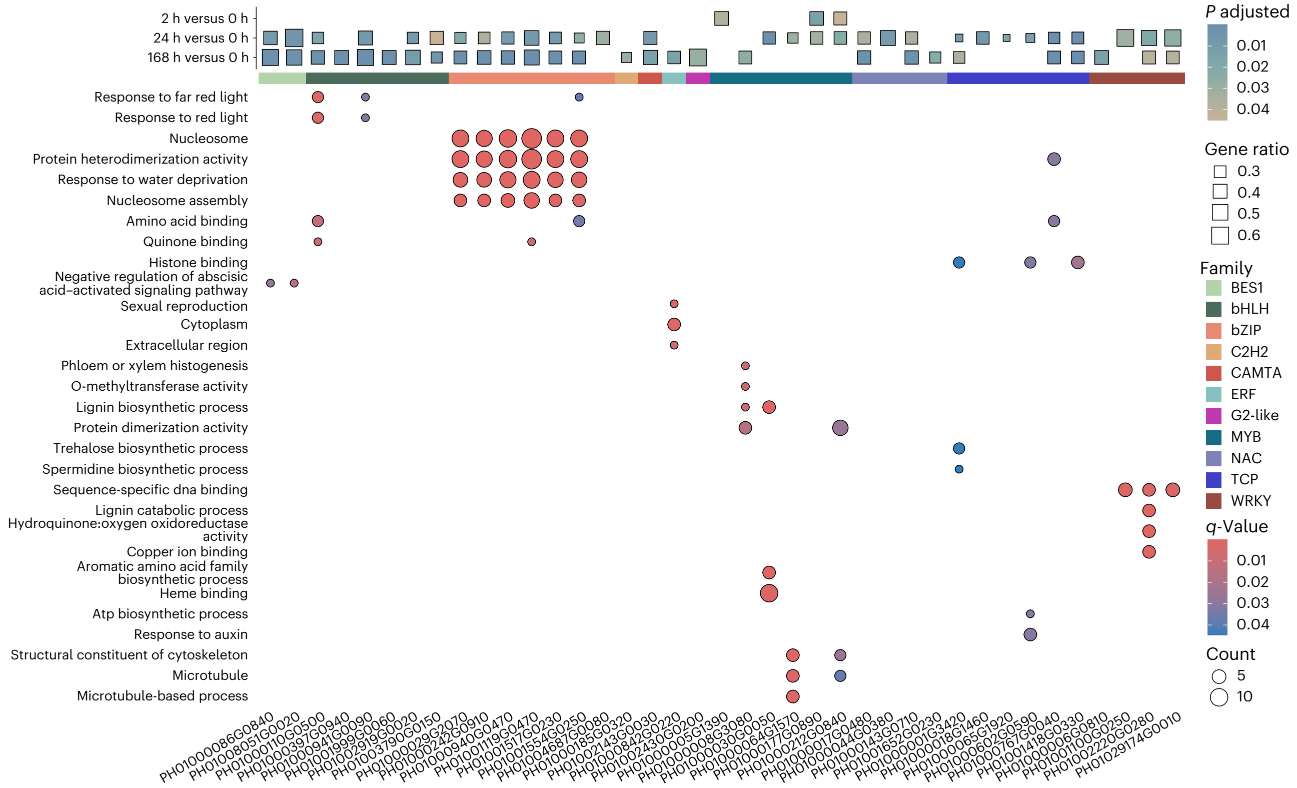


**Fig. 3 | Characterization of the biological functions of transcription factors involved in the response to cold stress in bamboo.** The upper side shows the transcription factors perturbed at different timepoints, which are obtained through GSEA analysis. The middle ribbon represents different transcription factor families. The lower side characterizes the biological functions of these perturbed transcription factors through ORA analysis. Both the upper and lower parts of the plot are analyzed by the comareCluster function and visualized by the dotplot function.

## Procedure 3: single-cell transcriptomic cell type annotation

● **TIMING** ~167.11 s

▲ **CRITICAL** Refer to Supplementary Note 2 for a version of the procedure where the syntax of the source code for the procedures is highlighted.

▲ **CRITICAL** The computational time for Procedure 3 is 167.11 s.

### Setup the environment and data objects

● **TIMING** ~9.92 s

1. Load the R packages into the R environment:

```
library(Seurat)
library(CelliD)
library(clusterProfiler)
library(ggplot2)
library(ggrepel)
library(ggsc)
```

2. Import example PBMC data:

```
pbmc_counts <- Read10X(data.dir = "hg19")
```

```
pbmc <- CreateSeuratObject(counts = pbmc_counts, project = "pbmc3k",
 min.cells = 3)
```

The Read10X function reads the output of the cellranger pipeline provided by 10X genomics and returns a unique molecular identified count matrix, which was subsequently used to create a Seurat object[65]. Genes expressed in less than three cells are removed. The object serves as a container that contains both data and analysis results.

## Data preprocessing workflow

● **TIMING ~6.61 s**

3. Quality control by removing unwanted cells:

```
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
pbmc <- subset(pbmc,
 subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5
)
```

The first command calculates the percentage of mitochondrial genes and the second command filters cells that express genes less than 200 or over 2,500, or the proportion of mitochondrial genes is over 5.

4. Normalize the data:

```
pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize")
pbmc <- ScaleData(pbmc)
```

The first command normalized the feature expression matrix for each cell by the total expression using the log-transform method. The second command applies *z*-score normalization to prevent the influence of highly expressed genes when performing dimensional reduction.

## Dimensionality reduction

● **TIMING ~25.95 s**

5. Identify highly variable genes:

```
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst",
 nfeatures = 2000)
```

The command applies the variance stabilizing transformation (vst) method to the count data and identifies the top 2,000 highly variable features.

6. Perform dimensional reduction by typing:

```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
pbmc <- RunUMAP(pbmc, dims = 1:10)
```

The first command performs PCA on the scaled data of the highly variable genes that are calculated in the previous step. The second command subsequently applies UMAP on the first ten PCs to improve visualization and interpretation of the data.

## Cluster cells and identify markers of cell clusters

● **TIMING ~110.04 s**

7. Construct a *K*-nearest neighbor graph:

```
pbmc <- FindNeighbors(pbmc, dims = 1:10)
```

8. Cluster cells into different groups:

```
pbmc <- FindClusters(pbmc, resolution = 0.5)
```

9. Find cluster biomarkers

```
pbmc <- RunMCA(pbmc)
cluster_markers <- GetGroupGeneSet(X = pbmc, n.features = 20)
```

The first command runs multiple correspondence analysis (MCA) to represent both cells and genes in the same space. The second command extracts the top 20 genes that are closest to each of the cell clusters.

### Cell type annotation

● TIMING ~14.59 s

10. Import cell marker gene set:

```
cell_marker_db <- read.gmt("c8.all.v2023.1.Hs.symbols.gmt")
```

11. Perform cell type enrichment analysis by typing:

```
cell_type_enrich_result <- compareCluster(cluster_markers,
 fun = "enricher", TERM2GENE = cell_marker_db
)
```

This command uses the compareCluster function to perform ORA analysis for each of the cell clusters using marker gene lists. The function was first published in 2012[7] for comparing biological themes among gene clusters obtained from different conditions. To our knowledge, it is the first implementation to allow such comparison and has been proven to be very effective for characterizing functional profiles of different clusters in single-cell studies[46].

12. Predict cell type:

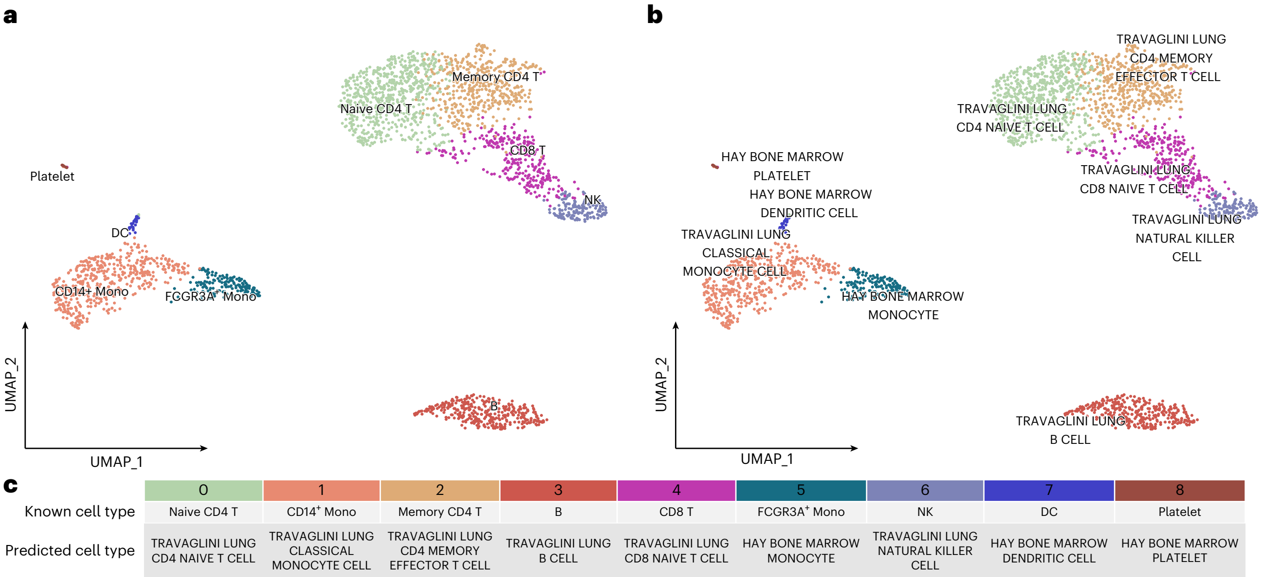
```
predict_cell_type <- function(enrich_result) {
 enrich_result <- as.data.frame(enrich_result)
 result <- split(
 enrich_result, enrich_result$Cluster
) |>
 vapply(function(x) {
 x$ID[which.min(x$p.adjust)]
 }, FUN.VALUE = character(1))
 cell_type <- gsub("_", " ", result) |>
 yulab.utils::str_wrap(18)
 names(cell_type) <- names(result)
 return(cell_type)
}
cell_type_predict <- predict_cell_type(cell_type_enrich_result)
```

In this study, we define the function 'predict_cell_type' to identify enriched cell types within each cluster. It assigns cell type identity with the lowest adjusted *P* value, which corrects for multiple hypothesis testing to minimize false discoveries.

◆ TROUBLESHOOTING

13. Assign cell type identity to clusters

```
pbmc <- RenameIdents(pbmc, cell_type_predict)
```



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Known cell type | Naive CD4 T | CD14$^+$ Mono | Memory CD4 T | B | CD8 T | FCGR3A$^+$ Mono | NK | DC | Platelet |
| Predicted cell type | TRAVAGLINI LUNG CD4 NAIVE T CELL | TRAVAGLINI LUNG CLASSICAL MONOCYTE CELL | TRAVAGLINI LUNG CD4 MEMORY EFFECTOR T CELL | TRAVAGLINI LUNG B CELL | TRAVAGLINI LUNG CD8 NAIVE T CELL | HAY BONE MARROW MONOCYTE | TRAVAGLINI LUNG NATURAL KILLER CELL | HAY BONE MARROW DENDRITIC CELL | HAY BONE MARROW PLATELET |

**Fig. 4 | Identify cell type at single-cell level. a**,**b**, UMAP plot of PBMC data with clusters labeled by known cell types (matched through canonical markers) (**a**) and predicted cell types (obtained through clusterProfiler analysis) (**b**). **c**, The table presents the correspondence between known and predicted cell types, which aids in better comparison.

14. Visualize predicted cell type identity (Fig. 4b):

```
cols <- c(’#B3D3AA’, ’#E88A71’, ’#DEAB76’, ’#CD574D’,
 ’#BF38AE’, ’#176D84’, ’#7D83B7’, ’#4040C6’, ’#994B41’)
sc_dim(pbmc) +
 sc_dim_geom_label(geom = ggrepel::geom_text_repel,
 color = "black", bg.color = "white") +
 scale_color_discrete(type=cols) +
 theme(legend.position = "none")
```

We utilize the in-house developed package, ggsc, to visualize the single-cell data using umap and label gene clusters. The ggsc package allows the use of the grammar of graphics syntax to visualize single-cell and spatial transcriptomic data.

## Troubleshooting

Troubleshooting advice can be found in Table 2.

**Table 2 | Troubleshooting**

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| Procedure 1, Step 2 | Error in compareCluster(geneClusters = genelist, fun = “enrichKEGG”, organism = “ko”): No gene can be mapped | The enrichKEGG function requires an internet connection to download KO data for enrichment analysis | Check that your running environment has a stable and reliable network connection |
| | | The input gene ID may not match the KEGG annotation ID | Check the input gene ID to make sure it is in the correct format, which should be a K number ID (e.g., ‘K00004’ or ‘K00007’); ensure that there are no unnecessary characters such as spaces in the ID |

**Table 2 (continued) | Troubleshooting**

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| Procedure 2, Step 9 | The number of groups displayed on the dot plot is less than the number of groups analyzed (in this example, the groups represent different timepoints) | During this step, not all groups exhibit enriched terms when subjected to the specified filtering criteria. It is normal for some groups not to show significant enrichment analysis results | If you wish to explore further, consider lowering the level of significance |
| Procedure 3, Step 12 | Error in vapply(split(enrich_result, enrich_result$Cluster), function(x) {: values must be length 1, but FUN(X[[1]]) result is length 0 | This error indicates that some cell clusters lack significant cell type annotations | Please verify whether the corresponding cell clusters are overly heterogeneous, or assess the coverage of the marker gene annotation |

## Timing

Running this protocol will take ~5 min on a laptop (purchased in 2017) with a dual-core processor and 4 GB of random-access memory. The running time can vary among different computers, and using a computer with better performance can substantially reduce the running time.

### Procedure 1: metabolomics and metagenomics functional enrichment analysis: ~89.26 s

Steps 1–2, Setup the environment and data objects: ~12.42 s
Steps 3–4, Metagenomic data differential analysis: ~41.16 s
Steps 5–6, Functional analysis of differential microbial genes: ~22.47 s
Steps 7–8, Metabolomic data differential analysis: ~5.75 s
Steps 9–10, Functional analysis of differential metabolites: ~7.43 s

### Procedure 2: transcription factor analysis pertaining to cold tolerance in PE: ~83.64 s

Steps 1–2, Prepare the expression matrix of Moso bamboo sequencing: ~0.93 s
Steps 3–4, Calculate log2FC using DESeq2: ~24.52 s
Steps 5–6, Extract log2FC, and sort genes based on the log2FC values: ~1.95 s
Steps 7–9, Transcription factor enrichment analysis to identify perturbed transcription factors at different timepoints: ~27.33 s
Steps 10–13, Predict the biological functions possibly regulated by the perturbed transcription factors: ~27.48 s
Steps 14–16, Visualization of transcription factor family annotation information: ~0.04 s
Step 17, All-in-one integration to reveal transcription factor perturbation and subsequent biological effects: 1.33 s

### Procedure 3: single-cell transcriptomic cell type annotation: ~167.11 s

Steps 1–2, Setup the environment and data objects: ~9.92 s
Steps 3–4, Data preprocessing workflow: ~6.61 s
Steps 5–6, Dimensionality reduction: ~25.95 s
Steps 7–9, Cluster cells and identify markers of cell clusters: ~110.04 s
Steps 10–14, Cell type annotation: ~14.59 s

## Anticipated results

### Procedure 1: microbiota–metabolome disease analysis

Although intrinsically related, CD and UC, two subtypes of IBD, exhibit distinct variations in their etiological mechanisms, affected anatomical sites, clinical presentations, therapeutic paradigms, prospective complications and long-term prognoses[66]. To delve deeper into the similarities and distinctions between these subtypes at the microbiome genetic and metabolic pathway levels, we executed a two-step comprehensive analysis. The results after following the steps in Procedure 1 are shown in Fig. 2.

Figure 2a shows the microbiome genetic landscape for CD and UC compared with normal controls. The software separates pathways that are only enriched in one disease from those that are enriched in both. Differences in the GeneRatio and adjusted *P* values are visualized using size and color scales. Empirical studies have provided robust evidence for perturbations in the metabolic derivatives associated with the tricarboxylic acid cycle, amino acids and lipidomic pathways in the IBD cohort. The identification of pathways enriched in both CD and UC, as illustrated in Fig. 2a, serves as a starting point for deeper investigation. The enrichment of a particular pathway suggests potential biological significance and warrants further targeted research to elucidate its role in disease pathology.

Differential pathway enrichments were also discerned between UC and CD. Specifically, UC manifested unique enrichment in pathways tropane, piperidine and pyridine alkaloid biosynthesis and biosynthesis of amino acids, as well as other pathways. Conversely, CD delineated a distinctive metabolic signature with elevated prominence in pathways including, but not limited to, arginine and proline metabolism, and beta-alanine metabolism[67].

Metabolically shared pathway enrichments across both subtypes encompassed bile acid metabolic dysregulation, perturbations in fatty acid metabolism and a hitherto underexplored ferroptosis pathway (Fig. 2b). Distinctively, cholesterol metabolism emerged as a salient feature exclusive to CD, whereas UC exhibited metabolic specificities in linoleic acid metabolism and diabetic cardiomyopathy.

Comparing the results, the arginine and proline metabolism pathway emerged as an axis of convergence across both omics spectrums, indicating its important role in IBD's etiological framework. This observation aligns with various studies that have reported the presence of altered arginine metabolism and arginine deficiency in IBD patients. Supplementation with arginine and the modulation of arginine metabolic pathways hold obvious promise as viable strategies for treating IBD[68].

### Procedure 2: analysis of cold-resistant transcription factors in plants

PE, commonly known as Moso bamboo, plays a critical role in the ecosystem due to its rapid growth, which contributes to carbon sequestration and oxygen release, thereby significantly mitigating the greenhouse effect. However, extreme cold conditions can negatively affect its growth and productivity. In this context, a comprehensive understanding of the dynamic transcriptional response of Moso bamboo to cold stress is crucial for devising effective strategies to safeguard its ecological functions. In this study, we relied on experimental data from Nie et al. wherein Moso bamboo was subjected to cold stress for durations of 0, 2, 24 and 168 h[45]. The results after following Procedure 2 are shown in Fig. 3.

Using the DESeq2 software, we calculated log2FCvalues of the transcriptomic data across various treatment groups (2 h versus 0 h, 24 h versus 0 h and 168 h versus 0 h) and applied the GSEA method to uncover key transcription factors activated at different stages. Using GSEA in this procedure, we have identified a series of transcription factor families significantly associated with cold response, with prominent members from the ERF, bHLH, bZIP, C2H2, CAMTA, MYB, NAC and WRKY families (Fig. 3). This discovery aligns with previous research, which has highlighted the crucial role these families play in plant cold response mechanisms[69]. The gene IDs of the transcription factors are listed at the bottom of Fig. 3. They are organized into different families and displayed with different color ribbons. The expression pattern for these transcription factors at different timepoints is shown using squares where the size and colour indicate the GeneRatio and the adjusted *P* value, respectively. From this, it is possible to see clear differences in expression patterns with time. Initially, at the 2 h timepoint, the analysis identified a modest set of transcription factors responding to cold stress, with only three being notably activated. As the exposure extended to 24 and 168 h, we observed a broader activation of cold-responsive transcription factors, including one member of the ERF family that was perturbed at 168 h. This is consistent with existing literature[70,71], which suggests a vital role for the ERF family in prolonged cold response. This phenomenon may suggest that plants progressively enhance their cold adaptation mechanisms, requiring a broader array of transcription factors to be involved in this biological regulation process.

To elucidate the downstream effects of these activated transcription factors during cold stress, we employed GO enrichment analysis to identify target biological processes that were regulated. These assist in exploring the biological processes associated with genes regulated by disrupted transcription factors during cold exposure. The biological processes significantly enriched with transcription factor target genes are visualized through the dot plot at the bottom of Fig. 3 with color highlighted by the $q$-value, which measures the proportion of false positives incurred when that particular test is called significant. The results revealed that a group of perturbed transcription factors primarily coordinates specific signaling pathways associated with cold stress response, particularly the 'response to water deprivation' and 'protein heterodimerization activity' pathways. These findings illuminate the intricacies of plant cellular responses under the adversities of cold environmental conditions, particularly during the critical phases of frost formation, where cells may traverse a gamut of physiological stresses inclusive of membrane structural aberrations and cellular dehydration phenomena. In a concerted effort to mitigate these external biological adversities, plants enact a dynamic adjustment of their transcription factor expression patterns, thereby facilitating a refined orchestration of gene expression and functional attributes within these pivotal signaling pathways[72]. Moreover, our detailed analysis delineated that the lignin synthesis pathway, a fundamental constituent of the cell wall, is notably enriched by the target genes under the aegis of these transcription factors. This datum substantiates the conjecture that during Moso bamboo's physiological adaptation to cold stimuli, lignin potentially assumes a critically important functional role, a proposition corroborated by existing research data.

In conclusion, the activation of these transcription factors appears to promote the strengthening of cold response in bamboo, which is crucial for its survival in low-temperature environments. Therefore, in combination with the existing transcription factor regulation databases, we can delve deeper into the functional mechanisms and regulation networks of transcription factors under various physiological conditions.

### Procedure 3: single-cell transcriptomics cell type annotation

A publicly available single-cell transcriptomics dataset was processed using Procedure 3. In this example dataset, the identity of each cell was determined using canonical markers.

The question was whether clusterProfiler could accurately classify different cell types using the transcriptomics data alone. The output for these analyses is shown in Fig. 4. Figure 4a,b show the classification of cell types based on canonical data and clusterProfiler predictions, respectively. In Fig. 4a, the cell labels come from the canonical data; in Fig. 4b, the cell labels were generated by clusterProfiler. Figure 4c compares the two annotations. Encouragingly, the cell type labels from both methods largely converge, underscoring the potential of using clusterProfiler in the realm of cell type annotation.

## Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

## Data availability

For the integrated analysis of metabolomics and metagenomics, the original metagenomic gene expression data and corresponding metadata, along with metabolomic metabolite expression profiles, were obtained from the supplementary materials of ref. 44. The data can be accessed through PubMed Central at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6342642. For the transcriptomics analysis of PE, we obtained the raw FASTQ sequencing data from the CNGBdb database (https://db.cngb.org/search/project/CNP0002243/) and conducted alignment and quantification analyses to determine the expression levels of individual genes. In Procedure 3, we acquired Illumina NextSeq 500 sequencing data for 2,700 PBMCs from 10X Genomics (https://cf.10xgenomics.com/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz). Source data are provided with this paper.

## Code availability

The original data, processed data and source code, including those for processing the original data and demonstrated in the protocol, are all deposited in the GitHub repository, https://github.com/YuLab-SMU/clusterProfiler_protocol/.

Received: 26 October 2023; Accepted: 13 May 2024;
Published online: 17 July 2024

## References

1. Paczkowska, M. et al. Integrative pathway enrichment analysis of multivariate omics data. *Nat. Commun.* **11**, 735 (2020).
2. Boyle, E. I. et al. 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 (2004).
3. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA* **102**, 15545–15550 (2005).
4. Xie, C., Jauhari, S. & Mora, A. Popularity and performance of bioinformatics software: the case of gene set analysis. *BMC Bioinforma.* **22**, 191 (2021).
5. Liu, X., Xu, K., Tao, X., Bo, X. & Chang, C. EnrichMiner: a biologist-oriented web server for mining biological insights from functional enrichment analysis results. Preprint at *bioRxiv* https://doi.org/10.1101/2023.07.12.548786 (2023).
6. Zhao, K. & Rhee, S. Y. Interpreting omics data with pathway enrichment analysis. *Trends Genet.* **39**, 308–319 (2023).
7. Yu, G., Wang, L.-G., Han, Y. & He, Q.-Y. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* **16**, 284–287 (2012).
8. Wu, T. et al. clusterProfiler 4.0: a universal enrichment tool for interpreting omics data. *Innovation* **2**, 100141 (2021).
9. Ding, J. & Zhang, Y. Analysis of key GO terms and KEGG pathways associated with carcinogenic chemicals. *Comb. Chem. High. Throughput Screen.* **20**, 861–871 (2017).
10. Li, Z. et al. Prediction and analysis of retinoblastoma related genes through gene ontology and KEGG. *Biomed. Res. Int.* **2013**, 304029 (2013).
11. Morgan, M. *Sequences, Genomes, and Genes in R/Bioconductor* (2013); https://www.ebi.ac.uk/sites/ebi.ac.uk/files/content.ebi.ac.uk/materials/2013/131021_HTS/genesandgenomes.pdf
12. Abromeit, F., Fäth, C. & Glaser, L. Annohub–annotation metadata for linked data applications. In *Proc. 7th Workshop on Linked Data in Linguistics (LDL-2020)* 36–44 (2020).
13. Liberzon, A. et al. Molecular signatures database (MSigDB) 3.0. *Bioinformatics* **27**, 1739–1740 (2011).
14. Martens, M. et al. WikiPathways: connecting communities. *Nucleic Acids Res.* **49**, D613–D621 (2021).
15. Conesa, A. et al. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* **21**, 3674–3676 (2005).
16. Li, S. & Xu, J. KAAS: a keyword-aware attention abstractive summarization model for scientific articles. In *International Conference on Database Systems for Advanced Applications* 263–271 (Springer, 2022).
17. Yu, G. & He, Q.-Y. ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization. *Mol. Biosyst.* **12**, 477–479 (2016).
18. Yu, G. Using meshes for MeSH term enrichment and semantic analyses. *Bioinformatics* **34**, 3766–3767 (2018).
19. Yu, G., Wang, L.-G., Yan, G.-R. & He, Q.-Y. DOSE: an R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics* **31**, 608–609 (2014).
20. Yu, G. et al. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* **26**, 976–978 (2010).
21. Yu, G., Wang, L.-G. & He, Q.-Y. ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics* **31**, 2382–2383 (2015).
22. Wang, Q. et al. Exploring epigenomic datasets by ChIPseeker. *Curr. Protoc.* **2**, e585 (2022).
23. Wickham, H. et al. Welcome to the Tidyverse. *J. Open Source Softw.* **4**, 1686 (2019).
24. Yu, G. Enrichplot: visualization of functional enrichment result. *R Package Version 1* (2021).
25. Wickham, H. in *ggplot2* 189–201 (Springer, 2016).
26. Merico, D., Isserlin, R., Stueker, O., Emili, A. & Bader, G. D. Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. *PloS ONE* **5**, e13984 (2010).
27. Hoover, A. R. et al. Single-cell RNA sequencing reveals localized tumour ablation and intratumoural immunostimulant delivery potentiate T cell mediated tumour killing. *Clin. Transl. Med.* **12**, e937 (2022).
28. Tan, Z. et al. HSPB8 is a potential prognostic biomarker that correlates with immune cell infiltration in bladder cancer. *Front. Genet.* **13**, 804858 (2022).
29. Liu, J. et al. Eleven genes associated with progression and prognosis of endometrial cancer (EC) identified by comprehensive bioinformatics analysis. *Cancer Cell Int.* **19**, 136 (2019).
30. Kuleshov, M. V. et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.* **44**, W90–W97 (2016).
31. Zhang, B., Kirov, S. & Snoddy, J. WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res.* **33**, W741–W748 (2005).
32. Gennady Korotkevich et al. Fast gene set enrichment analysis. Preprint at *bioRxiv* https://doi.org/10.1101/060012 (2021).
33. Falcon, S. & Gentleman, R. Using GOstats to test gene lists for GO term association. *Bioinformatics* **23**, 257–258 (2007).
34. Kolberg, L., Raudvere, U., Kuzmin, I., Vilo, J. & Peterson, H. gprofiler2—an R package for gene list functional enrichment analysis and namespace conversion toolset g:Profiler. *F1000Research* **9**, ELIXIR-709 (2020).
35. Dennis, G. et al. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol.* **4**, R60 (2003).
36. Zhou, Y. et al. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nat. Commun.* **10**, 1523 (2019).
37. Treveil, A. et al. ViralLink: an integrated workflow to investigate the effect of SARS-CoV-2 on intracellular signalling and regulatory pathways. *PLOS Comput. Biol.* **17**, e1008685 (2021).
38. Jiang, A., Lehnert, K., You, L. & Snell, R. G. ICARUS, an interactive web server for single cell RNA-seq analysis. *Nucleic Acids Res.* **50**, W427–W433 (2022).
39. Liu, J., Erenpreisa, J. & Sikora, E. Polyploid giant cancer cells: an emerging new field of cancer biology. *Semin. Cancer Biol.* **81**, 1–4 (2022).
40. Cui, G. et al. A carbon–nitrogen negative feedback loop underlies the repeated evolution of cnidarian–Symbiodiniaceae symbioses. *Nat. Commun.* **14**, 6949 (2023).
41. Nie, M. et al. Evolutionary metabolic landscape from preneoplasia to invasive lung adenocarcinoma. *Nat. Commun.* **12**, 6479 (2021).
42. Xu, S. et al. MicrobiotaProcess: a comprehensive R package for deep mining microbiome. *Innovation* **4**, 100388 (2023).
43. Chen, A. et al. Single-cell spatial transcriptome reveals cell-type organization in the macaque cortex. *Cell* **186**, 3726–3743.e24 (2023).
44. Franzosa, E. A. et al. Gut microbiome structure and metabolic activity in inflammatory bowel disease. *Nat. Microbiol.* **4**, 293–305 (2019).
45. Nie, Y. et al. Innovations and stepwise evolution of CBFs/DREB1s and their regulatory networks in angiosperms. *J. Integr. Plant Biol.* **64**, 2111–2125 (2022).
46. Vandereyken, K., Sifrim, A., Thienpont, B. & Voet, T. Methods and applications for single-cell and spatial multi-omics. *Nat. Rev. Genet.* **24**, 494–515 (2023).
47. Castanza, A. S. et al. Extending support for mouse data in the Molecular Signatures Database (MSigDB). *Nat. Methods* **20**, 1619–1620 (2023).
48. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2009).
49. Fernandes, A. D., Macklaim, J. M., Linn, T. G., Reid, G. & Gloor, G. B. ANOVA-like differential expression (ALDEx) analysis for mixed population RNA-seq. *PLoS ONE* **8**, e67019 (2013).
50. Kanehisa, M. & Goto, S. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
51. Wishart, D. S. et al. HMDB: the human metabolome database. *Nucleic Acids Res.* **35**, D521–D526 (2007).
52. Jewison, T. et al. SMPDB 2.0: big improvements to the small molecule pathway database. *Nucleic Acids Res.* **42**, D478–D484 (2013).
53. Jin, J. et al. PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. *Nucleic Acids Res.* **45**, D1040–D1045 (2016).
54. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
55. Ritchie, M. E. et al. *limma* powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **43**, e47 (2015).

56. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
57. McCarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33**, 1179–1186 (2017).
58. Lun, A., McCarthy, D. & Marioni, J. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research* **5**, 2122 (2016).
59. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
60. Cortal, A., Martignetti, L., Six, E. & Rausell, A. Gene signature extraction and cell identity recognition at the single-cell level with Cell-ID. *Nat. Biotechnol.* **39**, 1095–1102 (2021).
61. Wadi, L., Meyer, M., Weiser, J., Stein, L. D. & Reimand, J. Impact of outdated gene annotations on pathway enrichment analysis. *Nat. Methods* **13**, 705–706 (2016).
62. Pritykin, Y., Ghersi, D. & Singh, M. Genome-wide detection and analysis of multifunctional genes. *PLoS Comput. Biol.* **11**, e1004467 (2015).
63. Tian, F., Yang, D.-C., Meng, Y.-Q., Jin, J. & Gao, G. PlantRegMap: charting functional regulatory maps in plants. *Nucleic Acids Res.* **48**, D1104–D1113 (2019).
64. Anders, S. et al. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nat. Protoc.* **8**, 1765–1786 (2013).
65. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
66. Buie, M. J. et al. Global hospitalization trends for Crohn's Disease and ulcerative colitis in the 21st century: a systematic review with temporal analyses. *Clin. Gastroenterol. Hepatol.* **21**, 2211–2221 (2023).
67. Scoville, E. A. et al. Alterations in lipid, amino acid, and energy metabolism distinguish Crohn's disease from ulcerative colitis and control subjects by serum metabolomic profiling. *Metabolomics* **14**, 17 (2018).
68. Duboc, H. et al. Connecting dysbiosis, bile-acid dysmetabolism and gut inflammation in inflammatory bowel diseases. *Gut* **62**, 531 (2013).
69. Moura, J. C. M. S., Bonine, C. A. V., De Oliveira Fernandes Viana, J., Dornelas, M. C. & Mazzafera, P. Abiotic and biotic stresses and changes in the lignin content and composition in plants. *J. Integr. Plant Biol.* **52**, 360–376 (2010).
70. Lv, K. et al. Overexpression of an AP2/ERF family gene, BpERF13, in birch enhances cold tolerance through upregulating CBF genes and mitigating reactive oxygen species. *Plant Sci.* **292**, 110375 (2020).
71. Guo, Z. et al. Genome-wide analysis of the rhododendron AP2/ERF gene family: identification and expression profiles in response to cold, salt and drought stress. *Plants* **12**, 994 (2023).
72. Ding, Y. & Yang, S. Surviving and thriving: how plants perceive and respond to temperature stress. *Dev. Cell* **57**, 947–958 (2022).
73. Chen, S., Zhou, Y., Chen, Y. & Gu, J. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* **34**, i884–i890 (2018).
74. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).
75. Truong, D. T. et al. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nat. Methods* **12**, 902–903 (2015).
76. Franzosa, E. A. et al. Species-level functional profiling of metagenomes and metatranscriptomes. *Nat. Methods* **15**, 962–968 (2018).
77. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014).
78. Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat: discovering splice junctions with RNA-seq. *Bioinformatics* **25**, 1105–1111 (2009).
79. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
80. Kim, D., Paggi, J. M., Park, C., Bennett, C. & Salzberg, S. L. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat. Biotechnol.* **37**, 907–915 (2019).
81. Ghosh, S. & Chan, C.-K. K. in *Plant Bioinformatics* (ed. Edwards, D.) vol. 1374, 339–361 (Springer, 2016).
82. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinforma.* **12**, 323 (2011).
83. Liao, Y., Smyth, G. K. & Shi, W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* **30**, 923–930 (2014).
84. Anders, S., Pyl, P. T. & Huber, W. HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics* **31**, 166–169 (2015).
85. Consortium, G. O. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.* **32**, D258–D261 (2004).
86. Schriml, L. M. et al. Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Res.* **40**, D940–D946 (2011).
87. Piñero, J. et al. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res.* **45**, D833–D839 (2016).
88. Repana, D. et al. The Network of Cancer Genes (NCG): a comprehensive catalogue of known and candidate cancer genes from cancer sequencing screens. *Genome Biol.* **20**, 1 (2019).
89. Fabregat, A. et al. The reactome pathway knowledgebase. *Nucleic Acids Res.* **46**, D649–D655 (2017).

## Acknowledgements

This study was supported by the National Natural Science Foundation of China (32270677). We appreciate the feedback and support from clusterProfiler users.

## Author contributions

S.X., E.H. and Y.C. wrote the main manuscript and discussed the cases. S.X. and E.H. improved the code. Z.X. and X.L. conducted the pipeline and analyzed the results. L.Z., W.T., Q.W. and B.L. edited the paper for improvement. R.W., W.X., T.W. and L.X. reviewed the paper. G.Y. supervised the project, conducted the analysis and wrote the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41596-024-01020-z.

**Correspondence and requests for materials** should be addressed to Guangchuang Yu.

**Peer review information** *Nature Protocols* thanks Juri Reimand, Jianguo Xia and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Related links

**Key references using this protocol**

Yu, G. et al. *OMICS* **16**, 284–287 (2012): https://doi.org/10.1089/omi.2011.0118
Wu, T. et al. *Innovation* **2**, 100141 (2021): https://doi.org/10.1016/j.xinn.2021.100141
Ne, M. et al. *Nat. Commun.* **12**, 6479 (2021): https://doi.org/10.1038/s41467-021-26685-y
Alexandre, P. A. et al. *Genome Biol.* **22**, 273 (2021): https://doi.org/10.1186/s13059-021-02489-7
Sankowski, R. et al. *Nat. Med.* **30**, 186–198 (2024): https://doi.org/10.1038/s41591-023-02673-1

nature portfolio

Corresponding author(s): Guangchuang Yu

Last updated by author(s): Apr 30, 2024

# Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

## Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

| n/a | Confirmed | |
|---|---|---|
| ☒ | ☐ | The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement |
| ☒ | ☐ | A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly |
| ☒ | ☐ | The statistical test(s) used AND whether they are one- or two-sided<br>*Only common tests should be described solely by name; describe more complex techniques in the Methods section.* |
| ☒ | ☐ | A description of all covariates tested |
| ☒ | ☐ | A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons |
| ☒ | ☐ | A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals) |
| ☒ | ☐ | For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted<br>*Give P values as exact values whenever suitable.* |
| ☒ | ☐ | For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings |
| ☒ | ☐ | For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes |
| ☒ | ☐ | Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated |

*Our web collection on statistics for biologists contains articles on many of the points above.*

## Software and code

Policy information about availability of computer code

| | |
|---|---|
| Data collection | clusterProfiler |
| Data analysis | clusterProfiler |

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

## Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All datasets in the protocol are available from https://yulab-smu.top/clusterProfiler_protocol/

## Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

| | |
|---|---|
| Reporting on sex and gender | Not relevant |
| Population characteristics | Not relevant |
| Recruitment | Not relevant |
| Ethics oversight | Not relevant |

Note that full information on the approval of the study protocol must also be provided in the manuscript.

# Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

# Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| | |
|---|---|
| Sample size | All datasets used in the protocol are from the public domains. |
| Data exclusions | NA |
| Replication | NA |
| Randomization | NA |
| Blinding | NA |

# Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

## Materials & experimental systems

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ Antibodies |
| ☒ | ☐ Eukaryotic cell lines |
| ☒ | ☐ Palaeontology and archaeology |
| ☒ | ☐ Animals and other organisms |
| ☒ | ☐ Clinical data |
| ☒ | ☐ Dual use research of concern |

## Methods

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ ChIP-seq |
| ☒ | ☐ Flow cytometry |
| ☒ | ☐ MRI-based neuroimaging |