Received: 24 October 2023 | Revised: 3 January 2024 | Accepted: 8 January 2024

DOI: 10.1002/wics.1643

ADVANCED REVIEW

WIREs COMPUTATIONAL STATISTICS WILEY

From RNA sequencing measurements to the final results: A practical guide to navigating the choices and uncertainties of gene set analysis

Milena Wünsch[1,2] | **Christina Sauer**[1,2] | **Patrick Callahan**[1] | **Ludwig Christian Hinske**[3] | **Anne-Laure Boulesteix**[1,2]

[1]Institute for Medical Information Processing, Biometry, and Epidemiology, Faculty of Medicine, LMU Munich, Munich, Germany

[2]Munich Center for Machine Learning, Munich, Germany

[3]Institute for Digital Medicine, University Hospital of Augsburg, Augsburg, Germany

Correspondence
Milena Wünsch, Institute for Medical Information Processing, Biometry, and Epidemiology, Faculty of Medicine, LMU Munich, Marchioninistr. 15, 81377 Munich, Germany.
Email: milena.wuensch@ibe.med.uni-muenchen.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Numbers: BO3139/7-1, BO3139/9-1

Edited by: Emily Frieben, Commissioning Editor and David W Scott, Editor-in-Chief

Abstract

Gene set analysis (GSA), a popular approach for analyzing high-throughput gene expression data, aims to identify sets of related genes that show significantly enriched or depleted expression patterns between different conditions. In the last years, a multitude of methods have been developed for this task. However, clear guidance is lacking: choosing the right method is the first hurdle a researcher is confronted with. No less challenging than overcoming this so-called method uncertainty is the procedure of preprocessing, from knowing which steps are required to selecting a corresponding approach from the plethora of valid options to create the accepted input object (data preprocessing uncertainty), with clear guidance again being scarce. Here, we provide a practical guide through all steps required to conduct GSA, beginning with a concise overview of a selection of established methods, including Gene Set Enrichment Analysis and Database for Annotation, Visualization, and Integrated Discovery (`DAVID`). We thereby lay a special focus on reviewing and explaining the necessary preprocessing steps for each method under consideration (e.g., the necessity of a transformation of the RNA sequencing data)—an essential aspect that is typically paid only limited attention to in both existing reviews and applications. To raise awareness of the spectrum of uncertainties, our review is accompanied by an extensive overview of the literature on valid approaches for each step and illustrative R code demonstrating the complex analysis pipelines. It ends with a discussion and recommendations to both users and developers to ensure that the results of GSA are, despite the above-mentioned uncertainties, replicable and transparent.

This article is categorized under:
Statistical and Graphical Methods of Data Analysis > Analysis of High Dimensional Data
Data: Types and Structure > Data Preparation and Processing
Applications of Computational Statistics > Genomics/Proteomics/Genetics

KEYWORDS
data preprocessing uncertainty, gene set analysis, method uncertainty, multiplicity, parameter uncertainty

1 | INTRODUCTION

Gene set analysis (GSA) is a common approach to gaining insight into high-throughput gene expression data by detecting sets of related genes that show an enriched or depleted expression pattern between two conditions (such as cases and controls of a specific disease). GSA is thus considered an extension to the no less well-known differential expression analysis (Khatri et al., 2012), which, on the other hand, produces a list of individual genes that show a significant difference in gene expression between two conditions. Through the aggregation of genes with a common relation into gene sets, GSA boasts an increased statistical power and a simplified interpretation of the results, compared to differential expression analysis (Ackermann & Strimmer, 2009; Khatri et al., 2012).

Although a multitude of different methods have been proposed for GSA in the last years, it is hard for benchmark studies to keep pace with these developments. It has been observed that these benchmark studies only cover a small subset of available methods and that the results in terms of the best and poorest performers differ strongly (Xie et al., 2021). In particular, there are methods that score best in one benchmark study, while being the worst performer in another. This lack of reliable guidance for users aiming to choose the most suitable GSA method results in what we denote as *method uncertainty*, following the terminology introduced by Hoffmann et al. (2021) in a more general context.

This method uncertainty is accompanied by a variety of options within each individual GSA method, such as the choice of particular numeric parameters or the choice of the gene set database. These options in principle enable the methods' users to adapt them to the given research question and data at hand. While loose guidance is provided in the form of a default option for some parameters, the choice of the most suitable parameter value is often unclear, resulting in what we denote as *parameter uncertainty*.

The lack of guidance is even more pronounced when it comes to the generation of the required input objects of the individual GSA methods (e.g., list of differentially expressed genes, ranked list of all genes, or preprocessed gene expression data set). Here, the choice of the exact approach to preprocessing among a variety of valid options is largely left to the user, typically without any guidance through default approaches. Particularly for users with little bioinformatics experience, this so-called *data preprocessing uncertainty* may present a considerable difficulty when conducting GSA.

Although a variety of review articles addressing GSA have been published in the past, they primarily focus on the theoretical principles underlying the methods. For instance, some summarize the underlying structures of existing GSA methods into a common modular framework (Ackermann & Strimmer, 2009; Maleki et al., 2020). Others discuss the challenges and limitations of the three general approaches to GSA (Khatri et al., 2012). Practical guidance for GSA is provided for two web-based applications (Reimand et al., 2019), however, focusing primarily on using the applications and interpreting the results while touching only sparsely upon other choices related to data preprocessing.

In the general context of statistical analyses, the entirety of choices the user is confronted with is commonly referred to as *researchers' degrees of freedom* (Simmons et al., 2011). In the context of GSA, the researchers' degrees of freedom related to method, parameter, and data preprocessing uncertainty outlined above may make it difficult for users to obtain an overview of all steps required to conduct the analysis. This circumstance is further aggravated by the varying quality of the instructions and user manuals provided for the different methods. While for some methods, a detailed illustration of the required steps for preprocessing and running the method itself is provided, the manuals of other methods are considerably less instructive.

In view of this situation, the aim of this paper is to bring light into the darkness of all steps required to conduct GSA. While we cannot solve the spectrum of choices and uncertainties the user is confronted with by giving explicit

recommendations among the variety of options, we believe that the awareness of their existence and relevant characteristics is an important step towards more reliable results. The aim of our paper is threefold. First, we provide an overview of a selection of methods identified as widely used and/or well-performing in previous literature, including criteria guiding the choice (Sections 2 and 3). Second and most importantly, we review the practical steps related to preprocessing procedures required to generate the input objects accepted by the considered GSA methods (Section 4). This part is illustrated through documented example `R` codes. Third, we discuss the implications of the different types of uncertainties and formulate recommendations for methods' users and developers (Sections 5.2 and 5.3). In the Supporting Information, we address existing parameter and data preprocessing uncertainties by providing an extensive review of the multitude of available options and approaches.

2 | GENERAL PRINCIPLE

Methods developed for GSA typically have the same starting point and follow (variants of) a common framework. An understanding of this framework forms the cornerstone for recognizing similarities and differences between the methods and choosing the most suitable one for a specific research question at hand. In the following, we give an overview of the starting point of all GSA methods which is followed by a description of two general approaches to GSA.

Two components are required for GSA, namely (i) a gene expression data set, with corresponding assignments of the samples to the conditions, such as case and control of a specific disease, and (ii) a gene set database. The gene expression data set of dimension $N \times p$ contains the individual gene expression measurements of N genes across p samples in the form of integer count data. In principle, a higher value indicates a higher level of gene expression (see Section 3.1 in the Supporting Information for a more detailed description). The second component that is required for GSA is a gene set database which provides information on how the genes from the experiment are aggregated into gene sets. This aggregation is typically based on specific commonalities between the genes such as common chromosomal locations or biological functions (Subramanian et al., 2005). We note that individual genes can, dependent on the gene set database, be assigned to more than one gene set, resulting in a certain overlap between some gene sets.

Statistical methods for GSA can be divided into three major approaches, namely over-representation analysis (ORA), functional class scoring (FCS), and pathway topology (PT). These approaches differ in the extent to which the information from the gene expression data set is utilized and therefore greatly vary in the complexity of the underlying methodology.

In this work, we focus on methods ascribed either to ORA or FCS for which an overview of the underlying methodology is provided in Figure 1. This focus is substantiated by the selection criterion described in Section 3. However, for completeness, a short description of methods assigned to PT is provided in Section 1.2 in the Supporting Information.

2.1 | Over representation analysis

GSA methods classified as ORA (Draghici et al., 2003), which can be found on the left part of Figure 1, commonly are the least complex among the three approaches. The required input typically consists of a list of differentially expressed genes which is generated using a suitable method for differential expression analysis (Step 1). Then, a contingency table is generated which displays the frequencies of differential expression (differentially expressed versus not differentially expressed) and membership to the given gene set (member of the given gene set versus not a member of the given gene set) for all genes from the experiment (Step 2). For an illustration of the corresponding contingency table, see Table 1.

The null hypothesis states that the given gene set is not differentially enriched, i.e. that there is no association between membership of a given gene set and differential expression. Accordingly, a specific gene set is detected as differentially enriched if the number of its gene set members that are differentially expressed (H) is too high to be caused solely by chance (Step 3). The null distribution of H is modeled using the hypergeometric distribution. In the context of ORA, the corresponding population size (i.e., the entirety of genes) is typically referred to as "universe" or "set of background genes". Note that an elaboration on important aspects of the universe is provided in Section 4.1.2 in the Supporting Information. Eventually, a p-value of over-representation, on which the assessment of differential enrichment of the given gene set is based, is typically obtained using Fisher's exact test, which is described in further detail in Section 1.1 in the Supporting Information.

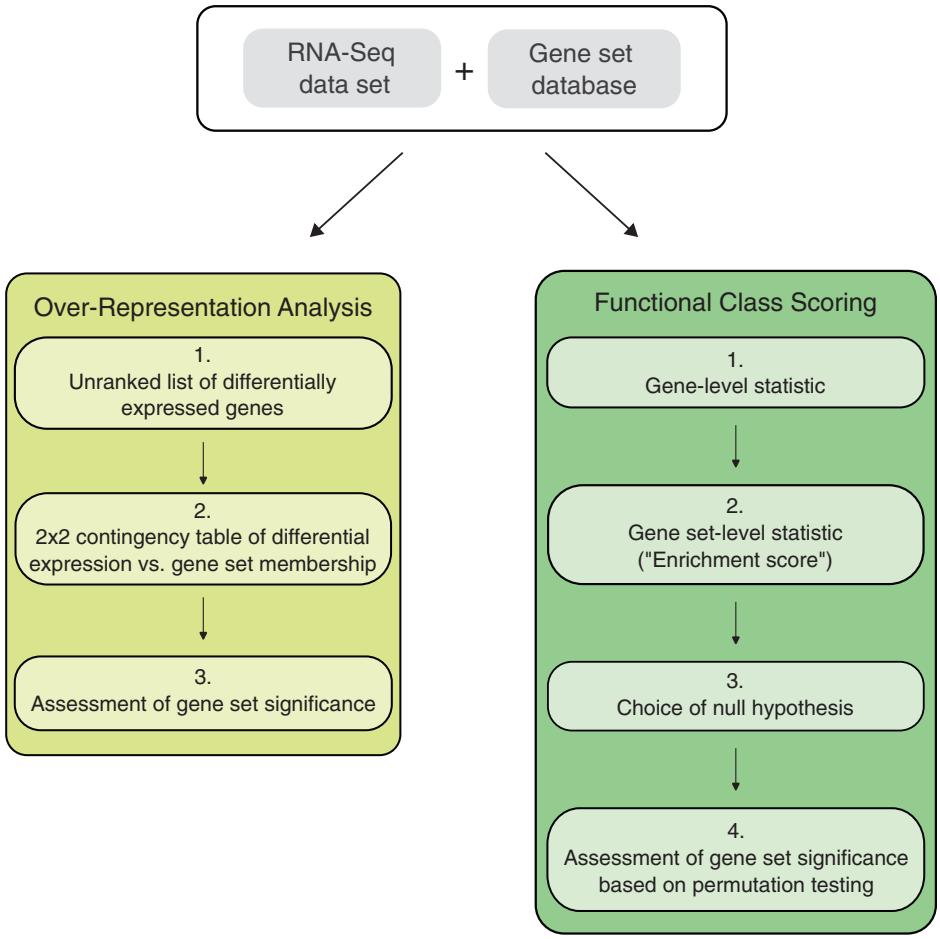

FIGURE 1 General overview of GSA approaches ORA and FCS, their required input, and preparatory steps. Depending on the form of the gene expression data set and practical aspects of the corresponding methods, additional preparatory steps might be necessary. FCS, functional class scoring; GSA, gene set analysis; ORA, over-representation analysis.

TABLE 1 Contingency table in over-representation analysis generated from all genes in the experiment.

	Differentially expressed	Not differentially expressed	Total
Member of gene set	H	$G-H$	G
Not a member of gene set	$L-H$	$N-L-(G-H)$	$N-G$
Total	L	$N-L$	N

Note: N indicates total number of genes from the experiment; H indicates number of genes from the input that are members of the given gene set ("hits"); L indicates number of genes in the input list; G indicates number of genes that are members of the given gene set.

2.2 | Functional class scoring

Unlike ORA, methods classified as FCS (see right part of Figure 1) do not only consider the subset of differentially expressed genes but instead utilize the expression profiles of the entirety of genes from the experiment (Maleki et al., 2020). For each gene, a gene-level statistic is generated which reflects the extent to which its expression pattern differs between both conditions (Step 1) (Ackermann & Strimmer, 2009). Here, a positive and negative value indicate an association with the first and second condition, respectively. A corresponding ranking of all genes from the experiment is generated based on their values of the gene-level statistic. A position at the upper and lower tail of the ranking indicates a strong association of the corresponding gene with the first and second conditions,

respectively, while a location in the middle implies that there is no particular association with any of the conditions. While some FCS methods generate this gene ranking internally, meaning that they only require the input of a (preprocessed) gene expression data set, others require the user to provide a gene ranking that has been generated externally.

Then, for a given gene set, the values of the gene-level statistics (or in some cases, the corresponding ranks) of all gene set members are aggregated into a gene set-level statistic (typically called enrichment score) to summarize whether the gene set members' expression patterns gravitate towards one of both conditions, that is, the genes are located towards the upper or lower end of the ranking, or else not associated with either of the conditions (Step 2) (Subramanian et al., 2005).

There are two types of null hypotheses that determine how to assess differential enrichment of a given gene set (Step 3), namely competitive and self-contained (Goeman & Bühlmann, 2007). Depending on the type of the null hypothesis, a *p*-value is obtained through an appropriate procedure of permutation that empirically generates a null distribution of enrichment scores against which the true enrichment score is compared (Step 4). The choice of the competitive null hypothesis implies comparing the association between the given gene set and the conditions to the association between the remaining genes and the conditions. In this case, the null distribution of enrichment scores is in practice approximated by repeating the analysis a large number of times with randomly generated fictive gene sets from the entirety of genes in the experiment that are of the same size as the given one, a procedure denoted as gene set permutation. For each of these randomly generated gene sets, an enrichment score is obtained from the initial ranking of all genes.

In contrast, a self-contained null hypothesis focuses on the given gene set, regardless of the remaining genes in the experiment. Accordingly, the null distribution of enrichment scores is obtained by generating a large number of random permutations of the assignments of the conditions to the samples (phenotype permutation). For each of these permutations, a corresponding gene ranking is generated from which the enrichment score is extracted.

3 | METHODS

Over the past years, a multitude of methods have been proposed for both ORA and FCS which differ in the specifics to assess differential enrichment. Beyond the distinction of ORA versus FCS, these may include biological aspects considered to be relevant as well as the handling of characteristics inherent to RNA sequencing (RNA-Seq) data which might affect the results.

Note that in this paper, we explicitly distinguish between *theoretical* and *computational* GSA methods under the aspect that each theoretical method, initially described in a scientific publication, is implemented in one or more computational methods (also denoted as "application"), either exactly or with some modifications. For instance, the common *theoretical* method Gene Set Enrichment Analysis (GSEA) from the publication of Subramanian et al. (2005) has been implemented in several *computational* methods, namely directly in a web-based application but also, with slight variations to the original underlying methodology, in several `R` packages. When referring to both theoretical and computational methods, we use the overall term *methods*, correspondingly.

From the multitude of existing (theoretical and computational) methods, we have made a selection based on popularity that is presented in Table 2. This selection is derived from the work of Xie et al. (2021) who provide a comprehensive reference database of existing GSA methods together with a quantification of the respective popularity. For a more detailed description of the selection criterion, inspect Section 2.1 in the Supporting Information. Note that the resulting selection consists only of methods assigned to ORA or FCS since PT generally scores considerably lower in terms of popularity in the considered reference database.

In the following, we provide a compact summary of each theoretical and corresponding computational method among this selection and focus on their respective specifics. For each of these methods, we additionally illustrate the correct use in the form of `R` code. Furthermore, we address the parameter uncertainties by demonstrating in the illustrations how the corresponding parameters can be adapted in the code.

TABLE 2 Overview of the considered GSA methods, considering that each *theoretical* method ("Theoretical method") is implemented in one or more *computational* methods ("Implemented in comp. method(s)").

Theoretical method	ORA or FCS	Implemented in comp. method(s)	R or web	Selection criterion
ORA	ORA	DAVID	Web	Popularity
		clusterProfiler	R	Popularity
GOSeq	ORA	GOSeq	R	Popularity
Gene Set Enrichment Analysis	FCS	clusterProfiler	R	Popularity
		GSEA	Web	Popularity
		GSEAPreranked	Web	Popularity
PADOG	FCS	PADOG	R	Performance

Note: While the column "R or Web" indicates whether the associated computational GSA method is in the form of an R or web-based application, the column "Selection criterion" distinguishes between a high popularity or a high performance as the criterion used to select the method.
Abbreviations: DAVID, Database for Annotation, Visualization, and Integrated Discovery; FCS, functional class scoring; ORA, over-representation analysis; PADOG, pathway analysis with down-weighting of overlapping genes.

3.1 | Methods for ORA

3.1.1 | clusterProfiler's ORA

In contrast to Database for Annotation, Visualization, and Integrated Discovery (DAVID) and GOSeq, which apply some modifications to the general ORA framework (see Section 2.1), a direct implementation is offered by the R package clusterProfiler (Wu et al., 2021). Given the list of differentially expressed genes as input, a two-dimensional contingency table (see Table 1) is generated which displays the frequency of differentially and non-differentially expressed genes in one dimension and the frequency of genes that are members of the given gene sets and genes that are not in the second dimension. Eventually, the p-value of over-representation of a given gene set is assessed using the hypergeometric distribution and Fisher's exact test.

3.1.2 | Database for Annotation, Visualization, and Integrated Discovery

DAVID (Huang et al., 2009a, 2009b) is a collection of web tools developed to provide an understanding of the biological meaning behind lists of genes. Within this collection of tools, the functional annotation tool provides a conservative variant of the classical ORA framework presented above such that the (adjusted) p-value increases for each gene set. This adaption, called "EASE score" (Hosack et al., 2003), favors larger gene sets since these are considered more "robust" to variations in the method used for generating the input list of differentially expressed genes. In particular, the detection of over-representation of gene sets containing only a single gene is precluded completely.

3.1.3 | GOSeq

GOSeq (Young et al., 2010) is an ORA method that addresses the circumstance that for RNA-Seq data, the probability of detecting a gene set as differentially enriched increases as the gene set members increase in transcript length, even if they remain constant in their level of differential expression between both conditions (Oshlack & Wakefield, 2009). A more detailed description of this so-called "length bias" is provided in Section 2.3 in the Supporting Information.

GOSeq counteracts length bias by incorporating each gene's estimated probability of being detected as differentially expressed depending on the gene's transcript length in the form of a probability weighting function.

By default, Wallenius' non-central hypergeometric distribution (Wallenius, 1963), which employs an extension of the standard hypergeometric distribution, is used to assess differential enrichment. In this method, it is assumed that within the given gene set, all genes have the same probability of being drawn from the universe and that this probability differs from the common probability of all genes outside of this gene set.

3.2 | Methods for FCS

Before giving a short overview of each FCS method under consideration, we will elaborate on an important distinction between the computational FCS methods which has a relevant impact on the practical steps that have to be performed by the user. In addition, we elaborate on common flexible parameter choices for these computational methods (i.e., parameter uncertainties) in Section 4.2 in the Supporting Information.

3.2.1 | Distinction: FCS I versus FCS II

One substantial difference between different computational methods classified as FCS is the form of the input object. While some require a manually ranked gene list as input, others accept a (preprocessed) gene expression data set as a whole, from which the gene ranking is generated internally. Since this distinction leads to a noticeable difference in the required preprocessing, it is addressed in this work by the subdivision of the computational FCS methods (see Table 3) into FCS I, which contains all computational methods that accept as input the (preprocessed) gene expression data set, and FCS II, which is based on a gene ranking. Note that for FCS II, this gene ranking is typically generated by conducting differential expression analysis and applying a suitable gene-level statistic to the quantities in the corresponding results table (see Section 4.6).

Since in the transition from the gene expression data set to the gene ranking, the information of the conditions (i.e., phenotypes) of the samples is lost, phenotype permutation, which is accompanied by a self-contained null hypothesis, cannot be performed for FCS II methods. Therefore, only gene set permutation (and thus a competitive null hypothesis) can be used to assess the significance of a given gene set.

We underline that the distinction between FCS I and FCS II is only made between computational FCS methods (and not theoretical FCS methods). For instance, the *computational* method `GSEA` is classified as FCS I and `GSEAPreranked` as FCS II while both provide an implementation of the *theoretical* method GSEA. Important aspects to consider when choosing between computational FCS I and FCS II methods are provided in Section 2.4 in the Supporting Information.

3.2.2 | Gene Set Enrichment Analysis (theoretical method)

The theoretical method GSEA (Subramanian et al., 2005) is classified as FCS and starts with a gene ranking that is based on the individual genes' associations with both conditions.

For a given gene set, the enrichment score is obtained by going down the gene ranking step by step and successively increasing a mathematical term for each gene that is a member of the gene set, while decreasing the term for each gene that is not. Each increase is weighted by the respective gene's association with the conditions, causing genes at the top and bottom of the ranked gene list to contribute more strongly to the enrichment score. The enrichment score is then obtained as the maximum deviation from zero of this term. Note that a more detailed description of the computation of the enrichment score is provided in Section 2.5 in the Supporting Information. By default, significance is assessed by testing a self-contained null hypothesis.

TABLE 3 Distinction of computational FCS methods based on the form of the required input object.

FCS method (computational)	FCS I or FCS II?
`GSEA`	FCS I
`GSEAPreranked`	FCS II
`clusterProfiler`'s GSEA	FCS II
`PADOG`	FCS I

Note: While FCS I comprises those computational FCS methods that accept as input the gene expression data set as a whole, FCS II methods require a gene ranking that reflects the genes' magnitudes of differential expression between both conditions.
Abbreviation: FCS, functional class scoring; GSEA, Gene Set Enrichment Analysis; PADOG, pathway analysis with down-weighting of overlapping genes.

3.2.3 | Gene Set Enrichment Analysis (computational method)

The theoretical method GSEA (see previous section) is implemented in the web-based application `GSEA` (Mootha et al., 2003; Subramanian et al., 2005). This computational method requires the input of the gene expression data set as a whole and is therefore classified as FCS I. By default, a gene ranking that represents the association with both conditions is generated internally using the signal-to-noise ratio.

In accordance with the self-contained null hypothesis, the enrichment of a given gene set is evaluated by randomly generating 1000 phenotype permutations.

3.2.4 | GSEAPreranked

In addition, the web-based application `GSEA` (see previous section) offers the version `GSEAPreranked` (Mootha et al., 2003; Subramanian et al., 2005) for which a user inputs their own list of genes that has already been ranked by a suitable gene-level statistic of choice. This version, which is categorized as FCS II accordingly, is recommended over the traditional computational method `GSEA` if the gene-level statistics provided by the application do not suit the gene expression data at hand. Further important aspects when choosing between `GSEAPreranked` and `GSEA` can be obtained from Section 2.4 in the Supporting Information. In accordance with FCS II, `GSEAPreranked` assesses the significance of a given gene set using gene set permutation.

3.2.5 | Pathway analysis with down-weighting of overlapping genes

PADOG (Tarca et al., 2013), which is short for "Pathway analysis with down-weighting of overlapping genes", is a theoretical FCS method whose corresponding computational method `PADOG` is categorized as FCS I. It assigns a higher weight to those genes in the computation of the enrichment score that are gene set-specific, meaning that they are assigned to a few or even only a single gene set and might therefore indicate a true connection between the associated gene set and the conditions of interest.

After quantifying each gene's magnitude of differential expression using a moderated variant of the *t*-statistic (Smyth, 2004), a gene set's enrichment score is obtained as the weighted mean of the t-statistics of the corresponding gene set members, with each gene's weight depending on its frequency of gene set memberships. Eventually, the significance of a standardized version of the enrichment score is assessed by testing a self-contained null hypothesis through phenotype permutation.

3.2.6 | clusterProfiler's GSEA

The `R` package `clusterProfiler` implements a modification of the theoretical method GSEA (see Section 3.2.2). This computational method requires a list of the genes from the experiment ranked by their magnitudes of differential expression as input and is therefore classified as FCS II. From this gene ranking, an enrichment score is calculated analogously to `GSEA`. However, the difference to the theoretical method, in which the conditions of the samples are permuted a certain number of times, is that significance is assessed by generating an empirical null distribution of enrichment scores by permuting the gene labels within the ranked list (Yu et al., 2015). Note that this approach also differs from gene set permutation.

4 | PREPROCESSING AND PRACTICAL ASPECTS

In general, a user has to procure a high amount of information about the preprocessing steps required to generate the input object for the chosen computational GSA method and additional practical aspects that are important when running it. A clear overview of the necessary steps is complicated by a discrepancy in the quality of the instructions and user manuals between the different methods. Furthermore, there is commonly a lack of default procedures among the

wide variety of different approaches to serve as guidance for the user. The enormous amount of informal guidance provided in online communities that is based on personal experience, as opposed to evidence-based guidelines from scientific literature, illustrates this issue.

Another difficulty a user typically encounters is that the practical steps required to conduct GSA can differ greatly across different methods and particularly between the different approaches. Even for computational methods based on the same theoretical methods, there can be differences regarding the form of the required input object.

To offer the user a guide through this seemingly obscure procedure of preprocessing, we have summarized the steps required to generate the accepted input object for each of the computational methods from Table 2. In our guide, we assume a gene expression data set in the form of RNA-Seq data with the genes identified in the Ensembl ID format (Cunningham et al., 2022). In addition, a graphical illustration is provided in Figure 2. While we focus on the creation of the input objects for the different computational methods, we also discuss important practical aspects of their internal parameters, such as the gene set database. Note that in this section, we will (almost) exclusively refer to the *computational* GSA methods since preprocessing takes place in the practical (and not theoretical) context. An extensive review of the researchers' degrees of freedom in the individual steps of preprocessing, leading to data preprocessing uncertainty, is provided in Section 5 in the Supporting Information.

4.1 | Pre-filtering

One of the first steps in the preprocessing of the RNA-Seq data is pre-filtering which refers to the exclusion of lowly expressed genes. The reasons for this procedure are discussed particularly in the context of differential expression analysis. One reason behind pre-filtering is that lowly expressed genes are unlikely to be detected as differentially expressed from the start and that their exclusion leads to a reduction of the number of statistical tests (which equals one per gene) to be performed (Love et al., 2014). As a consequence, the loss of statistical power accompanied by the correction for multiple testing is alleviated. The authors of edgeR (Robinson et al., 2010) argue from a biological viewpoint and state that a certain minimum expression level of a gene is required for it to be of biological importance.

In the context of GSA, the exclusion of genes with a particularly low magnitude of counts in RNA-Seq data is substantiated by the high likelihood that these counts are erroneously mapped to the given gene, even though it is not actually expressed in any of the samples (Subramanian et al., 2005).

4.2 | Gene ID conversion and removal of duplicated gene IDs

A variety of formats to identify gene IDs in a gene expression data set exist, such as Ensembl gene ID (Cunningham et al., 2022), NCBI (Entrez) gene ID (Sayers et al., 2022) and HGNC (HUGO) gene symbols (Seal et al., 2023). Oftentimes, there is a discrepancy between the gene ID format in the given gene expression data set and the format(s) accepted by the chosen computational GSA method (see Section 3.2.1 in the Supporting Information for an overview of the accepted gene ID formats). In this case, the gene IDs of the gene expression data set must be converted to the required format.

Different conversion tools rarely lead to an identical conversion pattern of the genes and guidance on the optimal choice of a conversion tool among the many options is scarce. Moreover, for some gene IDs in the initial format, the chosen conversion tool may not provide any corresponding gene ID in the converted format. These genes are lost for subsequent analysis and the gene expression data set is reduced accordingly.

Furthermore, there is often an ambiguous mapping between two distinct gene ID formats within a single conversion tool which can lead to duplicated gene IDs in two ways. First, a gene ID of the initial format can be converted to several gene IDs in the required format. Second, multiple distinct gene IDs of the initial format can be converted to the same gene ID in the required format. The two cases are addressed in detail in Section 3.2.2 in the Supporting Information.

Both circumstances require manual handling, however, there is a lack of guidelines on an optimal manner of duplicate gene ID removal provided by scientific publications. More specific details on possible approaches to removing duplicated gene IDs are provided in Section 5.2 in the Supporting Information.

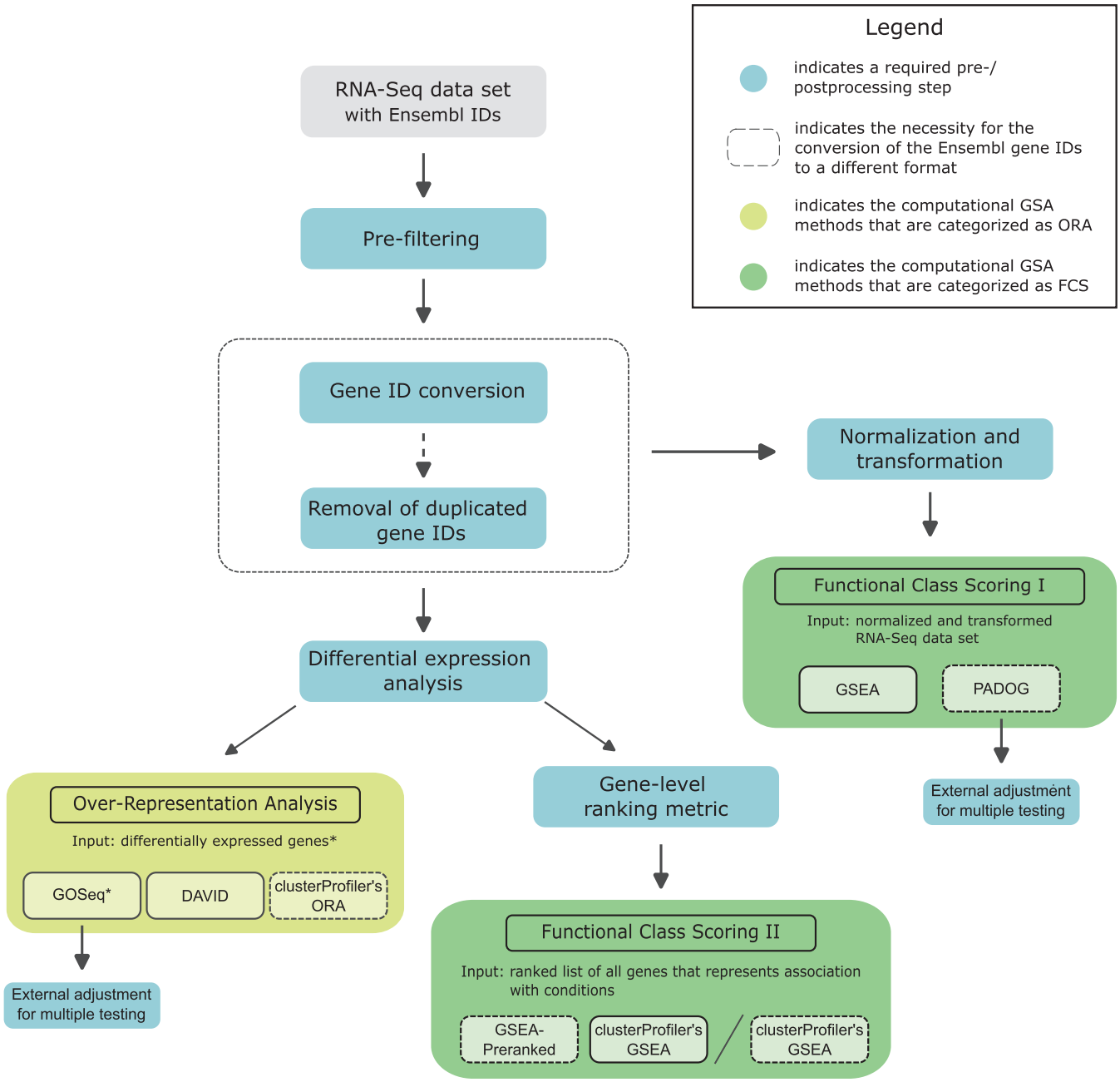

FIGURE 2 Overview of the practical GSA workflow for each computational GSA method, starting with an RNA-Seq data set with genes identified in the Ensembl ID format. A special focus is placed on the required preprocessing steps. The dashed borders indicate the necessity for a conversion of the gene IDs which is the case when the corresponding computational method does not accept the gene IDs in the Ensembl ID format. An overview of the accepted gene ID format(s) per computational GSA method is provided in Section 3.2.1 in the Supporting Information. Note that for `clusterProfiler`'s GSEA, a conversion to the NCBI (Entrez) ID format is required when using gene set database KEGG, while the initial Ensembl ID format can be retained for gene set database GO. The asterisk indicates that for `GOSeq`, the required input object consists of a named binary vector which indicates differential expression, as opposed to the list of differentially expressed genes. `GOSeq` and `PADOG` are the only two computational GSA methods from the selection for which the user has to perform multiple testing adjustment manually. GSA, gene set analysis; GSEA, Gene Set Enrichment Analysis; `PADOG`, pathway analysis with down-weighting of overlapping genes; RNA-Seq, RNA sequencing.

4.3 | Normalization of RNA-Seq data

In many cases, conducting GSA requires a suitable normalization of the RNA-Seq data set during preprocessing. Normalization aims to remove any sample-specific biases from RNA-Seq data that arise from the sequencing process itself

and would, unless accounted for, hinder an accurate comparison between different samples. An overview of the biases that are typically addressed by normalization methods, such as compositionality effects or differences in library size, is provided in Section 3.3 in the Supporting Information.

4.4 | Transformation of RNA-Seq data

In addition to normalization, a suitable transformation of the RNA-Seq data set is required if the chosen GSA method was specifically developed for data obtained using microarray technology. Note that this technology was the state-of-the-art technology to quantify gene expression before RNA-seq emerged. For instance, the web-based application `GSEA` and the `R` package `PADOG`, which are classified as FCS I, use the metrics signal-to-noise ratio and the moderated *t*-statistic, respectively, to generate the gene ranking. Such metrics typically assume continuous and particularly homoscedastic quantities, that is, (approximately) equal standard deviations of gene expression measurements across all magnitudes of quantified gene expression. While these assumed characteristics match those of microarray measurements, they do not hold for RNA-Seq data (Law et al., 2014). To address this issue, several methods have been proposed that aim to make microarray methods applicable to RNA-Seq data by addressing the discreteness and heteroscedasticity of these measurements with a suitable transformation (see Section 5.5 in the Supporting Information).

While in the context of differential expression analysis, the transformation of RNA-Seq data is particularly discussed by the authors of the method voom/limma (Law et al., 2014) (see Section 5.6.2 in the Supporting Information), the necessity of such transformation seems to be far less present in the context of GSA. For instance, in the user manual provided for the web-based application `GSEA`, it is stated that even though `GSEA` is commonly used for RNA-Seq data, its applicability to RNA-Seq measurements has not been fully investigated yet. This implies that `GSEA` is typically run on RNA-Seq data without any transformation to properly align its characteristics to microarray measurements. The only recommendation made is to perform a suitable method for normalization such as provided by DESeq2 (Love et al., 2014) which on its own is not sufficient to align the characteristics of RNA-Seq data to those of microarray measurements.

We note that a transformation of the RNA-Seq data can be bypassed for ORA and FCS II, whose required input objects are typically generated using differential expression analysis, by choosing a method for differential expression analysis specifically developed for RNA-Seq data (see Section 4.5).

4.5 | Differential expression analysis

The goal of differential expression analysis is to assess whether individual genes from the experiment exhibit different expression patterns between the two conditions. Typically, from the results of such analysis, the genes with an adjusted *p*-value below a certain threshold are classified as differentially expressed, whereas the remaining genes are categorized as not differentially expressed. As already described in Section 2.1, this dichotomous interpretation of differential expression analysis is used by ORA, which usually requires as input a list of differentially expressed genes. Note that, while ORA is typically performed for the full list of differentially expressed genes, it is sometimes run separately for the up- and down-regulated genes from this list; see Section 5.7 in the Supporting Information for further details.

In FCS, differential expression analysis is conducted in the course of preprocessing only for those computational methods that require as input a gene ranking generated by the user, that is, FCS II, while this ranking is created internally for FCS I. In the context of GSA, Reimand et al. (2019) recommend using methods for differential expression analysis that employ a variance stabilization, such as limma/voom (Law et al., 2014), DESeq2 (Love et al., 2014), or edgeR (Robinson et al., 2010).

4.6 | Gene-level statistic (for FCS II)

For FCS II methods, the required input list of all genes from the experiment that are ranked by their magnitude of differential expression is generated using a suitable gene-level statistic. This gene-level statistic is typically based on the quantities from the results table of the preceding differential expression analysis, such as the (un-adjusted) *p*-value and the estimated log fold change between both conditions. The gene-level statistic of each gene (and therefore the gene

ranking) is generated such that genes that show gene expression strongly in favor of one of both conditions have a high positive or negative value (and are placed at the upper or lower end of the ranking), respectively, while genes whose expression behavior does not change between both conditions have a value near zero and are located in the middle. For further information and suggestions on gene-level statistics, inspect Section 5.7 in the Supporting Information.

4.7 | Adjustment for multiple testing

In the context of GSA, a statistical test is performed for each gene set provided by the given gene set database, resulting in the necessity for multiple testing adjustment (for a more detailed explanation, refer to Section 5.8 in the Supporting Information). While the majority of computational GSA methods considered in this work perform multiple testing adjustment internally, `GOSeq` and `PADOG` constitute an exception. Both computational methods require multiple testing adjustment to be performed by the user, which can be done using the function `p.adjust()` from the `R` package `base`.

4.8 | Gene set database

As mentioned in Section 2, the gene set database provides information on how the genes from the experiment are aggregated into gene sets based on a specific relationship between the respective genes. Such a relationship could be a common chromosomal location or biological function (Subramanian et al., 2005). In the context of ORA, the gene set database additionally provides the set of background genes (i.e., universe) if the accepted input object consists of the list of differentially expressed genes so that the information about the entirety of genes from the experiment is lost, accordingly.

There are a variety of different gene set databases which typically differ in aspects such as the number of gene sets they contain (Mubeen et al., 2019). Mubeen et al. (2019) observe that in practice, the choice of the gene set databases is primarily made based on personal experience and preference, such as a tendency towards gene set databases that empirically yield preferable results, even though it should be made based on the underlying biological context.

4.9 | Specification of a seed for reproducibility (for FCS)

FCS methods, irrespective of the required input object, typically generate a large number of random permutations to empirically calculate a *p*-value of enrichment. To ensure that the same permutations are generated when the computational method is run at different points in time and thus lead to identical results, a seed can be specified in the form of an arbitrary integer number. The specification of a seed is therefore highly recommendable.

In contrast to FCS, ORA typically does not involve a random component as no random sampling or permuting is performed. Instead, enrichment is assessed based on parametric assumptions on the underlying null distributions. This also applies to the methods for differential expression analysis, resulting in no necessity and therefore no option to specify a seed.

4.10 | Visualization of the results

The GSA results are typically presented in the form of a ranking of the gene sets based on their adjusted *p*-values. For some methods, only a subset of all gene sets are displayed, such as those with an adjusted *p*-value below a prespecified threshold. Thereby, different options for adjusting the displayed set of gene sets often exist. However, users can also create more illustrative visualizations of the results using specially designed `Bioconductor` packages, for example, `enrichplot` (Yu, 2023). The corresponding visualizations additionally highlight specific important aspects, such as the genes involved in the differentially enriched gene sets. We refer to the associated vignettes for further details.

5 | UNCERTAINTIES, IMPLICATIONS, AND RECOMMENDATIONS

5.1 | Method, parameter, and data preprocessing uncertainty

The existence of method uncertainty becomes clear in Section 3, in which a variety of methods are presented. It is important to keep in mind that this selection is only a small subset of the plethora of methods available, a fact that further intensifies method uncertainty.

Even within an individual computational GSA method, there is typically a certain number of parameters (e.g., the gene set database) that can be modified to adapt the analysis strategy to the given research question and the data at hand, resulting in parameter uncertainty. Since the majority of flexible parameters differ between ORA and FCS, we elaborate on the common flexible parameter choices for both approaches separately (Sections 4.1 and 4.2 in the Supporting Information).

As already described, the flexibility offered between and within the individual GSA methods is accompanied by a variety of options in the preceding preprocessing for which guidance in the form of default approaches is scarce. Furthermore, in existing works on applied research, the exact details of the performed steps of preprocessing are typically neglected, if indicated at all. We address the corresponding data preprocessing uncertainty in the Supporting Information (see Section 5) by providing an overview of existing strategies from published literature for each step from Figure 2.

5.2 | Implications and recommendations for users

In the following, we provide additional implications and recommendations for users to navigate through the uncertainties inherent to GSA. For a concise summary, see Table 4.

Awareness of the uncertainties related to the use of GSA methods is the first step toward their correct use. In particular, the mere name of the method is not sufficient to characterize an analysis. A researcher, for example, cannot assume that they used the same analysis pipeline as used in a previous publication just because they used the same method. Attention should be devoted to all details of the implementation.

As far as possible, it is generally recommended to choose methods and parameters considering the biological setting and research question in the first place—as opposed to choices made after seeing the results as will be further discussed at the end of this section.

It is also important to note that, while some parameters can (or must) be chosen by the user to adapt to the underlying biological context and offer a certain amount of flexibility, others are to be considered as "technical" parameters related to the reproducibility of the results and computational resources. The latter thus do not offer such flexibility in the strict sense. In particular, the number of permutations in permutation-based methods should be set as large as

TABLE 4 Overview of the recommendations to users of GSA.

Recommendations to users of GSA
• Be aware of method, parameter, and data preprocessing uncertainty

• Report your analysis transparently
  ○ Include *all* choices regarding method, parameters, and data preprocessing approach
  ○ Make your code available
  ○ Report versions of all `R` packages and web-based applications used

• As far as possible: choose method, parameters, and data preprocessing approach *before starting your analysis*
  ○ Exception: sensitivity analysis to check the robustness of the results
  ○ Present *all* results in a structured way

• Technical parameters: do *not* offer true flexibility and must be specified *before* running the analysis
  ○ Random seed: *set a seed* for all methods in your analysis that contain a random component (such as random permutation) to ensure reproducibility
  ○ Number of permutations in permutation-based methods: set as high as computationally feasible or stick with the default value (if default is substantiated by the developers)

Abbreviation: GSA, gene set analysis.

computationally feasible. One should not run the analyses with different numbers of permutations but rather set this number to a large value in the first place. The seed—for methods involving a random component—is also not a parameter that should be specified flexibly. The seed should be, if at all, changed only to check whether results obtained with different seeds are very similar. It should particularly never be changed in the hope of obtaining "more satisfying results".

All changes of parameters or options conducted after seeing the results of interest, that is, a deliberate exploitation of the uncertainties in the GSA workflow, may tempt researchers to selective reporting. Typically, researchers will come back to their initial analysis and results if the change does not make the results more satisfying in their view. In contrast, they will adopt the changed pipeline if they prefer its results to those of the original pipeline. Such practice may seem natural at first glance so that many researchers who proceed this way may not be aware that this excessive fitting of the analysis strategy to the given data set is a form of cherry-picking, also denoted as fishing for significance in the context of statistical tests. In other contexts than GSA, it has been demonstrated that cherry-picking results in over-optimistic research findings that cannot be validated on new, independent data (Ullmann et al., 2023) and thus contributes to the so-called replication crisis. In this context, we want to emphasize that research findings that cannot be replicated on independent data are not valid. In the specific context of GSA, the extent to which over-optimistic results can be produced when method, parameter, and data preprocessing uncertainty are exploited has not yet been explored.

However, one may choose to apply different pipelines in the first place as a form of sensitivity analysis to check the robustness of the results. This amounts to the "report uncertainty strategy" suggested previously as one of many potential approaches to handling the multiplicity of analysis strategies (Hoffmann et al., 2021). The challenge will then be to present the results in a clearly arranged way to avoid confusion.

Irrespective of whether such robustness checks are performed or not, all choices done by researchers performing GSA should be transparently reported. In this context, code should be made available for the purpose of reproducibility, because all details of an analysis can barely be entirely reported in the form of text. Along with the code, researchers should always indicate the version of the chosen computational GSA method and the gene set database. In a recent survey on more than 200 gene set analyses, it was found that only a minority indicated the version of the software used while even less provided information on the version of the gene set database (Wijesooriya et al., 2022).

5.3 | Implications and recommendations for method developers

Considering the possibly important impact of uncertain choices on the results and the devastating effect of cherry-picking in terms of replicability, method developers should as far as possible provide practical guidance regarding parameter and data preprocessing uncertainty to future users. This could be done in the form of defaults and clear instructions regarding deviations from the defaults, which are ideally evidence-based rather than resulting from anecdotal experience.

Regarding the high degree of freedom affecting GSA, we regard neutral comparison studies (Boulesteix et al., 2017) investigating the behavior of methods dependent on parameters and preprocessing in different scenarios as playing a crucial role towards the more proficient use of these methods by users, ultimately leading to more reliable results in the field. In particular, keeping in mind that the "one method fits all data sets" philosophy does not sufficiently reflect the complexity of such analyses (Strobl & Leisch, 2024), we argue that comparison studies should investigate a variety of settings, data sets, and methods' variants in addition to the default parameters. Particularly, factorial comparison designs are used to identify which features of the methods or pipelines make a difference. Suppose as a simplified example of such a factorial comparison design that the authors of a method we call "A" apply it together with a particular metric for differential expression assessment, while the authors of another method called "B" use another metric. In this case, it may make sense to investigate the behavior of all 2×2 combinations of methods and metrics rather than sticking to the original two combinations, as already demonstrated in another context (Nießl et al., 2024).

6 | ILLUSTRATIONS IN R

Alongside this review, we provide an illustration of the practical steps to be performed in R for each step from Figure 2. For an overview of the structure of the illustration, refer to Section 6 in the Supporting Information. For this

illustration, the user has the choice to work through several R files (available on GitHub), however, they can also inspect them visually as a part of A Practical Guide through Running Gene Set Analysis in R.

AUTHOR CONTRIBUTIONS

Milena Wünsch: Conceptualization (supporting); formal analysis (lead); software (lead); visualization (lead); writing—original draft (lead); writing—review and editing (equal). **Christina Sauer:** Conceptualization (equal); writing—review and editing (equal). **Patrick Callahan:** Software (supporting); visualization (supporting). **Ludwig Christian Hinske:** Conceptualization (supporting); writing—review and editing (supporting). **Anne-Laure Boulesteix:** Conceptualization (equal); funding acquisition (lead); supervision (lead); writing—review and editing (equal).

ACKNOWLEDGMENT

The authors thank Savanna Ratky for language correction. Open Access funding enabled and organized by Projekt DEAL.

FUNDING INFORMATION

This work is supported in part by funds from the German Research Foundation (DFG: BO3139/7-1 and BO3139/9-1).

CONFLICT OF INTEREST STATEMENT

The authors have declared no conflicts of interest for this article.

OPEN RESEARCH BADGES

This article has earned an Open Data badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. The data are available at https://github.com/chillemille/RIllustrations.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Milena Wünsch https://orcid.org/0009-0001-1982-9260
Christina Sauer https://orcid.org/0000-0003-2425-7858
Patrick Callahan https://orcid.org/0000-0003-1769-7580
Ludwig Christian Hinske https://orcid.org/0000-0001-7273-5899
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947

RELATED WIREs ARTICLE

Gene expression modular analysis: an overview from the data mining perspective

REFERENCES

Ackermann, M., & Strimmer, K. (2009). A general modular framework for gene set enrichment analysis. *BioMed Central Bioinformatics*, *10*, 47.

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BioMed Central Medical Research Methodology*, *17*, 138.

Cunningham, F., Allen, J. E., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Austine-Orimoloye, O., Azov, A. G., Barnes, I., Bennett, R., Berry, A., Bhai, J., Bignell, A., Billis, K., Boddu, S., Brooks, L., Charkhchi, M., Cummins, C., da Rin Fioretto, L., ... Flicek, P. (2022). Ensembl 2022. *Nucleic Acids Research*, *50*(D1), D988–D995.

Draghici, S., Khatri, P., Martins, R. P., Ostermeier, G. C., & Krawetz, S. A. (2003). Global functional profiling of gene expression. *Genomics*, *81*(2), 98–104.

Goeman, J. J., & Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: Methodological issues. *Bioinformatics*, *23*(8), 980–987.

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., & Boulesteix, A. L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, *8*(4), 201925.

Hosack, D. A., Dennis, G., Sherman, B. T., Lane, H. C., & Lempicki, R. A. (2003). Identifying biological themes within lists of genes with ease. *Genome Biology*, *4*, R70.

Huang, D. W., Sherman, B. T., & Lempicki, R. A. (2009a). Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, *37*(1), 1–13.

Huang, D. W., Sherman, B. T., & Lempicki, R. A. (2009b). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, *4*(1), 44–57.

Khatri, P., Sirota, M., & Butte, A. J. (2012). Ten years of pathway analysis: Current approaches and outstanding challenges. *PLoS Computational Biology*, *8*(2), e1002375.

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). Voom: Precision weights unlock linear model analysis tools for RNA-Seq read counts. *Genome Biology*, *15*, R29.

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *Genome Biology*, *15*, 550.

Maleki, F., Ovens, K., Hogan, D. J., & Kusalik, A. J. (2020). Gene set analysis: Challenges, opportunities, and future research. *Frontiers in Genetics*, *11*, 654.

Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstråle, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., ... Groop, L. C. (2003). PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, *34*(3), 267–273.

Mubeen, S., Hoyt, C. T., Gemünd, A., Hofmann-Apitius, M., Fröhlich, H., & Domingo-Fernández, D. (2019). The impact of pathway database choice on statistical enrichment analysis and predictive modeling. *Frontiers in Genetics*, *10*, 1203.

Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2024). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *Biometrical Journal*, *66*, 2200238.

Oshlack, A., & Wakefield, M. J. (2009). Transcript length bias in RNA-Seq data confounds systems biology. *Biology Direct*, *4*, 1–10.

Reimand, J., Isserlin, R., Voisin, V., Kucera, M., Tannus-Lopes, C., Rostamianfar, A., Wadi, L., Meyer, M., Wong, J., Xu, C., Merico, D., & Bader, G. D. (2019). Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. *Nature Protocols*, *14*(2), 482–517.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140.

Sayers, E. W., Bolton, E. E., Brister, J. R., Canese, K., Chan, J., Comeau, D. C., Connor, R., Funk, K., Kelly, C., Kim, S., Madej, T., Marchler-Bauer, A., Lanczycki, C., Lathrop, S., Lu, Z., Thibaud-Nissen, F., Murphy, T., Phan, L., Skripchenko, Y., ... Sherry, S. T. (2022). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, *50*(D1), D20–D26.

Seal, R. L., Braschi, B., Gray, K., Jones, T. E. M., Tweedie, S., Haim-Vilmovsky, L., & Bruford, E. A. (2023). Genenames.org: The HGNC resources in 2023. *Nucleic Acids Research*, *51*(D1), D1003–D1009.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*, 3.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., & Mesirov, J. P. (2005). Gene Set Enrichment Analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, *102*(43), 15545–15550.

Strobl, C., & Leisch, F. (2024). Against the "one method fits all data sets" philosophy for comparison studies in methodological research. *Biometrical Journal*, *66*, 202200104.

Tarca, A. L., Bhatti, G., & Romero, R. (2013). A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. *PLoS One*, *8*(11), e79217.

Ullmann, T., Peschel, S., Finger, P., Müller, C. L., & Boulesteix, A. L. (2023). Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering. *PLoS Computational Biology*, *19*(1), e1010820.

Wallenius, K. T. (1963). Biased sampling: the non-central hypergeometricprobability distribution. PhD Thesis. Stanford University, Department of Statistics.

Wijesooriya, K., Jadaan, S. A., Perera, K. L., Kaur, T., & Ziemann, M. (2022). Urgent need for consistent standards in functional enrichment analysis. *PLoS Computational Biology*, *18*(3), e1009935.

Wu, T., Hu, E., Xu, S., Chen, M., Guo, P., Dai, Z., Feng, T., Zhou, L., Tang, W., Zhan, L., Fu, X., Liu, S., Bo, X., & Yu, G. (2021). clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *The Innovations*, *2*(3), 100141.

Xie, C., Jauhari, S., & Mora, A. (2021). Popularity and performance of bioinformatics software: The case of gene set analysis. *BioMed Central Bioinformatics*, *22*, 191.

Young, M. D., Wakefield, M. J., Smyth, G. K., & Oshlack, A. (2010). Gene ontology analysis for RNA-Seq: Accounting for selection bias. *Genome Biology*, *11*, R14.

Yu, G. (2023). enrichplot: Visualization of functional enrichment result. R package version 1.18.4.

Yu, G., Wang, L.-G., Yan, G.-R., & He, Q. Y. (2015). DOSE: An R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics*, *31*(4), 608–609.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Wünsch, M., Sauer, C., Callahan, P., Hinske, L. C., & Boulesteix, A.-L. (2024). From RNA sequencing measurements to the final results: A practical guide to navigating the choices and uncertainties of gene set analysis. *WIREs Computational Statistics*, *16*(1), e1643. https://doi.org/10.1002/wics.1643