Marini *et al. BMC Bioinformatics (2021) 22:610*
https://doi.org/10.1186/s12859-021-04461-5

BMC Bioinformatics

SOFTWARE

Open Access

GeneTonic: an R/Bioconductor package for streamlining the interpretation of RNA-seq data

Federico Marini[1,2*], Annekathrin Ludt[1], Jan Linke[1,2] and Konstantin Strauch[1]

*Correspondence:
marinif@uni-mainz.de
[2] Center for Thrombosis and Hemostasis (CTH), University Medical Center of the Johannes Gutenberg University Mainz, Langenbeckstr. 1, 55131 Mainz, Germany
Full list of author information is available at the end of the article

Abstract

Background: The interpretation of results from transcriptome profiling experiments via RNA sequencing (RNA-seq) can be a complex task, where the essential information is distributed among different tabular and list formats—normalized expression values, results from differential expression analysis, and results from functional enrichment analyses. A number of tools and databases are widely used for the purpose of identification of relevant functional patterns, yet often their contextualization within the data and results at hand is not straightforward, especially if these analytic components are not combined together efficiently.

Results: We developed the `GeneTonic` software package, which serves as a comprehensive toolkit for streamlining the interpretation of functional enrichment analyses, by fully leveraging the information of expression values in a differential expression context. `GeneTonic` is implemented in R and Shiny, leveraging packages that enable HTML-based interactive visualizations for executing drilldown tasks seamlessly, viewing the data at a level of increased detail. `GeneTonic` is integrated with the core classes of existing Bioconductor workflows, and can accept the output of many widely used tools for pathway analysis, making this approach applicable to a wide range of use cases. Users can effectively navigate interlinked components (otherwise available as flat text or spreadsheet tables), bookmark features of interest during the exploration sessions, and obtain at the end a tailored HTML report, thus combining the benefits of both interactivity and reproducibility.

Conclusion: `GeneTonic` is distributed as an R package in the Bioconductor project (https://bioconductor.org/packages/GeneTonic/) under the MIT license. Offering both bird's-eye views of the components of transcriptome data analysis and the detailed inspection of single genes, individual signatures, and their relationships, `GeneTonic` aims at simplifying the process of interpretation of complex and compelling RNA-seq datasets for many researchers with different expertise profiles.

Keywords: RNA-Seq, Functional enrichment analysis, Data interpretation, Interactive data analysis, Data visualization, Transcriptomics, R, Bioconductor, Shiny, Reproducible research

Background

In modern life and clinical sciences, RNA-sequencing (RNA-seq) is an essential tool for studying gene expression and its regulation [1]. High-throughput sequencing technologies generate readouts for a large number of molecular entities simultaneously, posing challenges to proper hypothesis generation and data interpretation [2]. Among the typical bioinformatic workflows, differential expression (DE) analysis is often employed to identify the genes showing evidence for statistically significant changes, thus being candidate effectors for regulation across the sampled experimental conditions [3].

Most studies where these techniques are being adopted result in a list containing tens to thousands of gene candidates, with their associated effect size and significance level—often reported as $\log_2$ fold change ($\log_2$FC) and adjusted p-values, respectively. Putting these results into biological context by leveraging existing knowledge is essential for facilitating the interpretation of data at a systemic level, and enabling novel discoveries [4].

Commonly used knowledge bases for the purpose of functional enrichment analysis include Gene Ontology (GO) [5, 6], KEGG [7, 8], REACTOME [9], and MSigDB [10, 11], where the genes are organized either in simple lists (gene sets, or signatures), or as pathways by accounting for the interactions occurring among the respective members; throughout this manuscript, we will use these terms interchangeably. The analysis at the functional level not only aims to reduce the complexity of high dimensional molecular data (grouping thousands of genes and proteins to just several hundreds of coherent entities), but also increasing the explanatory power of the underlying observed mechanisms [12].

A large variety of computational methods and software have been designed for functional enrichment analysis [13], and despite their different implementations, they can still be grouped in three main categories, as identified by Khatri [12]: (1) Over-Representation Analysis (ORA), contrasting only the set of DE genes against a background of expressed genes; (2) Functional Class Scoring (FCS), including e.g. Gene Set Enrichment Analysis (GSEA, [14]) and its different flavors, incorporating a feature-(gene-) level score/statistic, later aggregated at the pathway level to avoid the choice of a binary threshold; (3) Pathway Topology (PT) based approaches, which utilize the additional information of graph/network structure describing the interactions [15]. The most widely adopted approaches in this context are ORA and FCS methods, owing to their ease of applicability, fast runtime, and relevance of resulting gene set rankings, as shown in a recent benchmarking effort [16].

Visualization techniques are widely used to make sense of enrichment analysis results, where gene sets might also be highly redundant, thus making the prioritization and interpretation of interesting candidates more challenging [17, 18]. Numerous tools and applications aim to simplify the interpretation step by adopting a diverse range of methods and visual summaries, and these include BiNGO [19], ClueGO [20, 21], GOrilla [22], REVIGO [23], GOplot [24], AgriGO [25], NaviGO [26], WebGestalt [27], CirGO [28], AEGIS [29], FunSet [30], hypeR [31], KeggExp [32], Metascape [33], pathfindR [34], ShinyGO [35], ViSEAGO [36], STRING [37], GSOAP [38], GOMCL [39], and netGO [40].

Aggregating and summarizing the identified categories is an efficient way to capture and distill the main underlying biological aspects, exploiting visual methods that can efficiently encode the essential information of a table. Among the commonly used visualization methods, many apply different ways of grouping and displaying similar genes or gene sets together, including graph-like representations, clustered heatmaps (either genes by samples, or genesets by samples), or wordclouds. Good visualizations enable discovering underlying trends in the data in an unbiased fashion, and are essential components for the proper communication of results in interdisciplinary projects [18, 41].

Datasets and gene set collections increase constantly in their size and complexity, constituting a major barrier for the interpretability of transcriptomic data and their enrichment results, to the point that a potential bottleneck for omics data is the so-called *tertiary* analysis, opposed to mapping and quantification (*primary* analysis) and statistical testing (*secondary* analysis) [42]. Efficient platforms that enable advanced workflows for a wide range of users can play a big role in providing the required level of interactivity, while guaranteeing the adherence to gold standard methods and to best practices for reproducible analyses [43–45].

The different atomic elements for a typical RNA-seq analysis (expression table, results from differential expression, functional enrichment results) can stem from different pipeline outputs, yet they need to be combined together, e.g. in a report created following the rules of literate programming [46]. By providing accessible summaries with proper data visualization and interpretation methods, in formats that facilitate dynamic shareable outputs, such frameworks can greatly reduce the time to generate novel hypotheses and insight. Often, this task is not straightforward to carry out, as different software solutions or environments might be chosen, resulting in different file formats, thus increasing the difficulty for practitioners to explore all relevant aspects of the data at hand, even if common sets of gene and pathway identifiers are adopted.

A number of solutions have been developed in diverse languages (mostly R, Python, Java) to address the challenges listed above, but no software package provides a comprehensive framework for assisting the proper interpretation of RNA-seq data; interested readers can find a comparative overview of the features of the above mentioned tools in Additional file 1: Table S1.

Here we present `GeneTonic`, an R/Bioconductor package aiming to streamline the identification of relevant functional patterns, as well as their contextualization in the data and results at hand, by combining in a seamless way all the pieces of information relevant for a transcriptomic analysis. The `GeneTonic` package is composed by a Shiny web application, with a variety of standalone functions to perform the analysis both interactively as well as in a programmatic way. `GeneTonic` requires as input the results generated by each analytic step (quantification, DE testing, functional enrichment), which are usually shared as separate tables or spreadsheets by bioinformaticians and core facility service providers, in formats that are suitable to standardization.

`GeneTonic` makes it easy to generate visualizations, starting from bird's eye perspective summaries (gene-geneset graphs, enrichment maps, also linked to interactive tables in the web application), as well as getting in-depth dedicated summaries for each geneset of interest. User actions enable further insight and deliver additional information (e.g. gene info boxes, geneset summaries, and signature heatmaps), with drilldown tasks activated by simple mouse clicks. While simple operations within the call to the GeneTonic() main function makes the result set more interpretable, our package also supports built-in RMarkdown reporting as a foundation for computational reproducibility, to conclude an interactive exploration session [47, 48]. We carefully designed the user interface, enabling the required tasks in a straightforward way, as a result of an open and continuous dialogue with researchers adopting this tool in its early development. Users can learn-by-doing the functionality of GeneTonic via guided tours, creating a common ground for experimentalists and analysts to explore transcriptomic data at the desired depth and efficiently generate novel insights [49].

GeneTonic connects together a number of R/Bioconductor packages, implementing the current best practices in RNA-seq data analysis, and facilitates the communication between experts of different disciplines. Harmonizing the output of the many analysis steps, possibly performed also with a variety of approaches, GeneTonic is a powerful tool for digesting and enjoying any RNA-seq dataset: the interactivity is a compelling means to empower end users for the exploration of many features of interest, and by providing a report with full code snippets, we support analyses that are reproducible and easily extendable. The GeneTonic package is available at https://bioconductor.org/packages/GeneTonic/, and a public instance is available for demonstration purposes at http://shiny.imbei.uni-mainz.de:3838/GeneTonic.

Implementation

General design of GeneTonic

The GeneTonic package is written in the R programming language, leveraging many existing packages currently available in the Bioconductor project, which constitute the foundation for a broad spectrum of analytic workflows in computational biology and bioinformatics [50, 51], and the Shiny framework for interactivity [52]. The typical use case for GeneTonic expects researchers to run the web application locally, providing the atomic components of a typical RNA-seq analysis workflow (Fig. 1, top section).

GeneTonic is designed to be used after the main steps of DE and functional enrichment analyses have already been completed. While this might seem a limiting factor, we wanted to acknowledge that a plethora of validated methods for performing functional analyses at the pathway level exist, and similarly, well established statistical methods for DE are available [1, 16]. Our focus was rather on providing a standardized interface (via so-called shaker functions) to automatically handle the outputs of the different tools which most users might be familiar with, so that GeneTonic retains a wide applicability with respect to the upstream analysis workflows—this is illustrated in the use cases of Additional file 2 and Additional file 3, which include comprehensive reports covering exploratory data analysis, differential expression analysis, and functional enrichment analysis for two exemplary datasets [53, 54], documented in the repository https://github.com/federicomarini/GeneTonic_supplement.

The required input components are stored as in the DESeq2 workflow [55], using classes descending from the versatile SummarizedExperiment container, *de facto* the adopted standard for interoperability in the Bioconductor ecosystem [50]. Tabular information can be provided as simple data frame objects, either imported from textual output of the different tools, or converted internally by the shaker functions

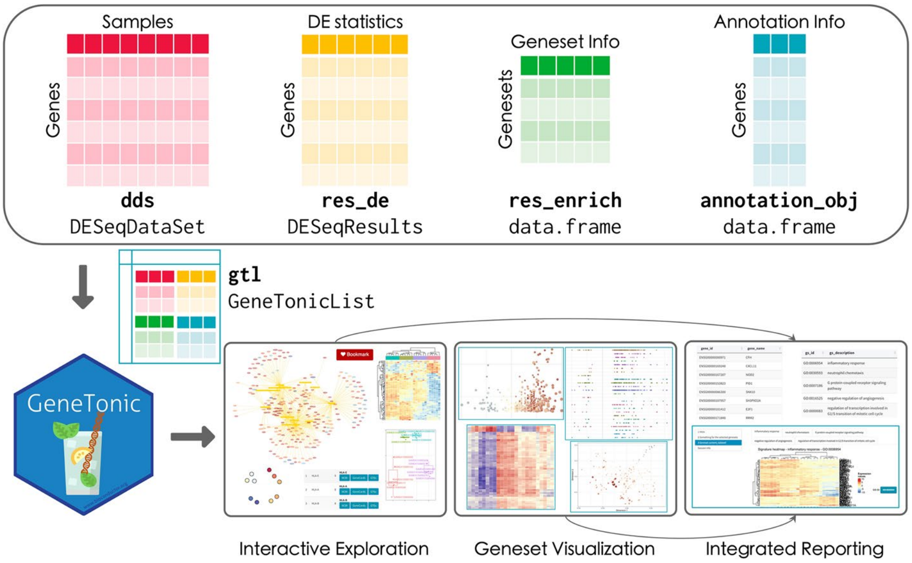


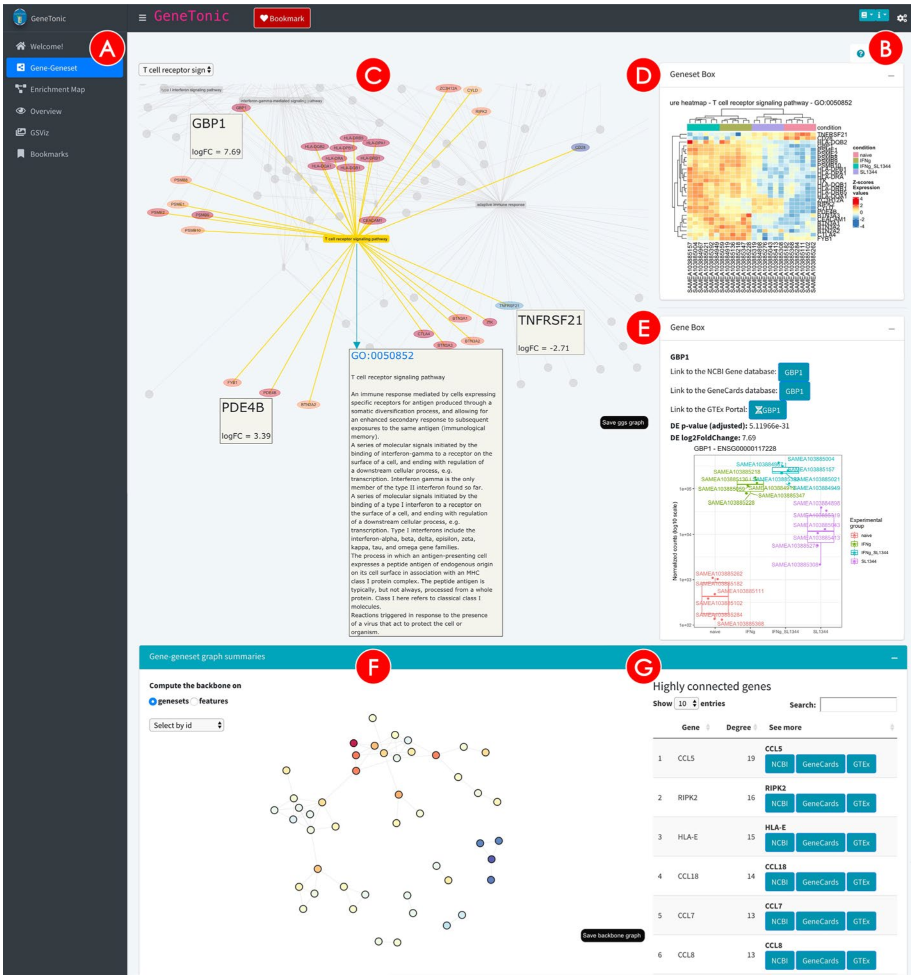
Fig. 1 Overview of the `GeneTonic` workflow. Four elements are required to optimally use the functionality of `GeneTonic` (top section): `dds`, a `DESeqDataSet`, with the information related to the expression values; `res_de`, a `DESeqResults` object, storing the results of the differential expression analysis; `res_enrich`, a data frame with the result from the enrichment analysis; `annotation_obj`, a data frame containing different sets of matched identifiers for the features of interest. These can be combined into a `GeneTonicList` object (middle section), which can be directly fed to `GeneTonic` and its functions, enabling (bottom section) interactive exploration and a variety of visualizations on the geneset enrichment results, both exploited to generate an integrated HTML report of the provided data

of `GeneTonic`. We encourage users to adopt stable feature identifiers, such as ENSEMBL or Gencode [56, 57], and enable the automated conversion to HGNC gene symbols via annotation tables.

Most of the implemented functionality can be accessed by a single call to the main `GeneTonic()` function, with the different visualizations and summaries directly available from the dedicated sections of the web application. These functions are also exported for usage in scripted analyses such as RMarkdown HTML reports, making it easy to automate tasks while still creating interactive widgets that can be explored in depth offline (Fig. 1, bottom section).

The user interface (shown in Fig. 2) is structured with the layout provided by the `bs4Dash` package [58], which implements Bootstrap 4 over the infrastructure of `shinydashboard` [59]. The main features include a sidebar menu (Fig. 2A) to navigate the different sections of the app, a header and a collapsible control panel to provide widgets which define the general behavior of the main panel, displayed in the dashboard body (Fig. 2C).

To instruct users on how to efficiently leverage the exploration components, we enhance the content provided in the use case vignette with guided tours of the interface (Fig. 2B), implemented via the `rintrojs` package [60]. This learning-by-doing paradigm invites the user to perform the actions that reflect typical usages in each module, and can be seen as a dynamic extension of the static documentation format.



GO:0050852

T cell receptor signaling pathway

An immune response mediated by cells expressing specific receptors for antigen produced through a somatic diversification process, and allowing for an enhanced secondary response to subsequent exposures to the same antigen (immunological memory).
A series of molecular signals initiated by the binding of interferon-gamma to a receptor on the surface of a cell, and ending with regulation of a downstream cellular process, e.g. transcription. Interferon gamma is the only member of the type II interferon found so far.
A series of molecular signals initiated by the binding of a type I interferon to a receptor on the surface of a cell, and ending with regulation of a downstream cellular process, e.g. transcription. Type I interferons include the interferon-alpha, beta, delta, episilon, zeta, kappa, tau, and omega gene families.
The process in which an antigen-presenting cell expresses a peptide antigen of endogenous origin on its cell surface in association with an MHC class I protein complex. The peptide antigen is typically, but not always, processed from a whole protein. Class I here refers to classical class I molecules.
Reactions triggered in response to the presence of a virus that act to protect the cell or organism.

GBP1

Link to the NCBI Gene database: GBP1

Link to the GeneCards database: GBP1

Link to the GTEx Portal: GBP1

DE p-value (adjusted): 5.11966e-31

DE log2FoldChange: 7.69

Compute the backbone on

● genesets ○ features

Select by id

Highly connected genes

Show 10 entries Search:

	Gene	Degree	See more
1	CCL5	19	CCL5 NCBI GeneCards GTEx
2	RIPK2	16	RIPK2 NCBI GeneCards GTEx
3	HLA-E	15	HLA-E NCBI GeneCards GTEx
4	CCL18	14	CCL18 NCBI GeneCards GTEx
5	CCL7	13	CCL7 NCBI GeneCards GTEx
6	CCL8	13	CCL8 NCBI GeneCards GTEx

Fig. 2 Screenshot of the Gene-Geneset panel in the `GeneTonic` application. The main navigation of the app is performed via the sidebar menu (**A**), while the options common to the different panels are available in the control bar (toggled with the cogs icon, close to the **B** circle). The main area of the *Gene-Geneset* panel (**C**) contains an interactive bipartite graph connecting genesets to their respective components, color coded according to the type and the expression changes. Upon clicking on any node, a Geneset Box (**D**) or a Gene Box (**E**) can be automatically generated for facilitating further exploration. The backbone of the graph above (**F**) represents a compact summary of the bipartite projections, and a table displays the most highly connected nodes (**G**), as potential hubs with regulatory roles in multiple genesets

Collapsible and tab-based elements allowed us to build a rich user interface, yet without adding too much visual clutter, which would hamper the usability of the analysis sessions—and by that reduce the ability to extract relevant insight.

All the required elements for running `GeneTonic` are provided at the beginning of the execution, meaning that the navigation throughout the different modules can take place with the usual iteration cycles that build up a full in-depth exploration. As this process can become time-consuming, we implemented a dedicated bookmarking system to temporarily store the genes and gene sets of interest, either by clicking on the dedicated button or with a keystroke (defaulting to the left control key). A summary for these selected features is automatically rendered in the *Bookmarks* section, where the user can generate a full report on the provided input parameters, focusing on the aspects picked up during the live session (Fig. 1, bottom). It is then easy to reconstruct and reproduce the analytic rationale, and share the rendered outputs with cooperation partners, or simply store them for the purpose of careful documentation.

Typical usage workflow

The typical session with `GeneTonic` can start once the required inputs are provided to the main function, as illustrated in Fig. 1.

In order to use `GeneTonic`, the following inputs are required: (1) `dds`, a `DESeqDataSet`, the main component in the `DESeq2` framework, storing the information related to the expression matrix; (2) `res_de`, encoded as `DESeqResults` for containing the results of the differential expression analysis; (3) `res_enrich`, i.e. the result from the enrichment analysis, likely converted through one of the `shaker` functions for preprocessing (or manually, if feeding this from a tool currently not supported), structured as a data frame with a minimal set of required variables (pathway identifier, description, significance level, and affected genes); (4) `annotation_obj`, the gene annotation data frame, i.e. a table with at least two columns, `gene_id` for a set of unambiguous identifiers (e.g. ENSEMBL ids), and `gene_name`, containing a human-readable set of names, e.g. HGNC-based gene symbols.

Conveniently, a single named list containing these inputs (Fig. 1, middle section) can be provided as an alternative format, with many functions of `GeneTonic` accepting a `gtl` parameter (standing for "`GeneTonicList`"). This simplifies the creation of context-dependent serialized objects that can be easily shared by data analysts to experimental collaborators. More details on the format of the components to be provided to the `GeneTonic()` main function can be found in the dedicated sections ("Data processing") of the use cases in Additional file 2 and Additional file 3.

In its current version, `GeneTonic` can directly handle the output of different tools, selected for being among the most commonly used in pathway analysis, including `topGO` [61], `clusterProfiler` [62], DAVID [63], Enrichr [64], g:Profiler [65, 66], and `fgsea` [67]—these are showcased in the code included in Additional file 2 and Additional file 3. We plan to extend the compatibility of `GeneTonic` with the output of newly developed tools, or alternatively welcome contributions on the project homepage on GitHub (https://github.com/federicomarini/GeneTonic).

All the components of the `GeneTonic()` application can be seamlessly used by leveraging sets of shared gene identifiers across the different input objects. This makes it possible to compute aggregate scores for each gene set, e.g. averaging the $\log_2$ fold change of all the affected members of the gene set, or computing a Z-score based on the standardized sum of the number of genes regulated in either direction. As gene sets cannot take into account the topological information of a pathway, this is a valid surrogate means to summarize the effect between conditions at the functional level, and can be visually encoded in the outputs of the `GeneTonic` dedicated routines.

The process of data exploration and interpretation is iterative by its own nature. `GeneTonic` supports this by employing a variety of visual summaries (gene-geneset bipartite graphs, enrichment maps, geneset volcano plots, and more in the dedicated app sections—Fig. 2 and Additional file 4: Fig. S1). We also offer methods to efficiently extract the most meaningful affected biological themes, e.g. by grouping similar categories and selecting a representative pathway for each subset, in order to simplify the redundancy often found in functional enrichment results. Whenever possible, we provide additional information boxes for genes and genesets (Fig. 2D, E) to facilitate drilldown tasks and better understand the whole data components of the project. A number of automatically generated action buttons link directly to external databases, such as AmiGO [6], NCBI [68], GeneCards [69], GTEx [70], enabling more in-depth analysis of particular genesets or genes, without the need to type all the entries of interest.

While the main way of using the functionality of `GeneTonic` is probably via its web application, we designed all the underlying functions to be able to handle standard objects and classes adopted by the current Bioconductor workflows, and therefore their output can be also incorporated in information-rich HTML reports and existing scripted analyses without additional effort. Indeed, the report itself created via the `happy_hour()` function is an exemplary RMarkdown document, which users can edit and extend as they see fit. Literate programming approaches were initially conceived by the seminal work of Knuth [46] and have been currently refined in the `knitr` framework [71] and in the Jupyter notebook system [72]. These techniques constitute a powerful toolkit to ensure the reproducibility of computational analyses [43, 45, 73]. The creation of such an HTML document is also intended as the recommended concluding step of a typical usage session for `GeneTonic`.

In case additional or bespoke visual representations of the input objects (e.g. MA plot for the DE results, customized heatmaps, ...) should be created, the `iSEE` Bioconductor package [74] can be used for this purpose. We provide a specific export function, combining the provided inputs into a `SummarizedExperiment` object that can be readily further explored in an instance of `iSEE`, by properly accessing the `assays`, `colData`, and `rowData` slots.

Results

In this section, we will illustrate the functionality of `GeneTonic`, showcasing the results for the analysis of a human RNA-seq dataset of macrophage immune stimulation, published in [53]. The data is made available via the `macrophage` Bioconductor package, which contains the files output from the Salmon quantification (version 0.12.0—[75]), against the Gencode v29 human reference. Expression values, summarized at the gene level, are available from 6 individual donors, in 4 different conditions. We will focus on the comparison between Interferon gamma treated samples versus naive samples—this scenario is also shown in the demonstration instance available at http://shiny.imbei.uni-mainz.de:3838/GeneTonic.

A comprehensive report on the processing for this dataset and its usage with `GeneTonic` is included in Additional file 2. Additional file 3 showcases the usage of `GeneTonic` on the findings of the work of [54] (A20-deficiency in mouse microglia cells), describing also an alternate entry point for running `GeneTonic` with objects from the `edgeR` workflow for differential expression [76].

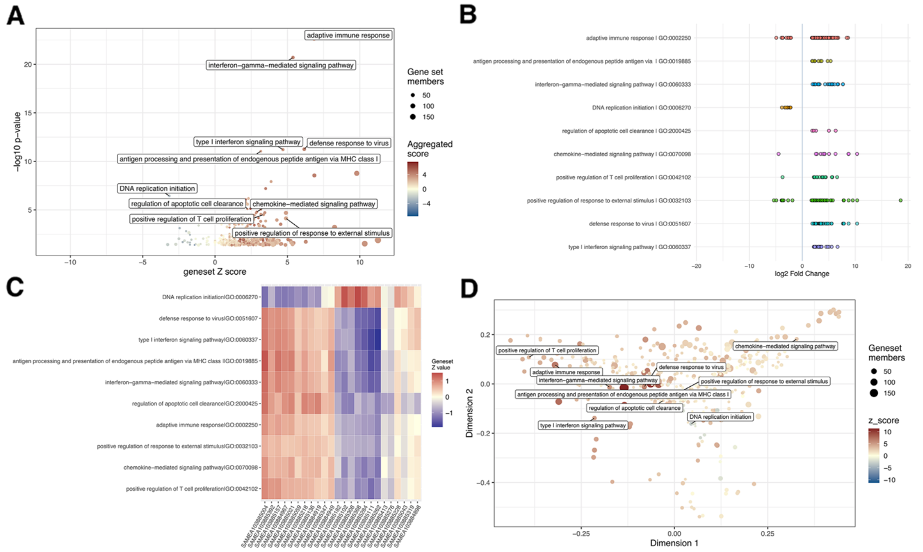


Fig. 3 Graphical summaries of enrichment results. **A** A geneset volcano plot (`gs_volcano()`), with the -$\log_{10}$ *p* value depicted against the geneset Z-score, with a subset of representative affected functions, generated from the output of the `gs_fuzzyclustering()` function. The size of the points maps the information of the geneset size. **B** An enhanced visual summary for the enrichment results, displaying the contributions of the single genes to each gene set (with the directionality as $\log_2$FC), created with `enhance_table()`. **C** A heatmap for the matrix of sample-wise geneset Z-scores for the same subset used in the other subpanels, generated with `gs_scoresheat()` on the output of `gs_scores()`. **D** A multidimensional scaling plot for genesets, colored by their Z-score, representing the set similarity. This can be generated with the `gs_mds()` function

Augmenting functional enrichment results with expression data

The majority of functions in `GeneTonic` requires only a minimal set of information on the pathways enrichment results, i.e. a gene set identifier, its description, and a measure of significance for the enrichment, often specified as a *p* value. Nevertheless, it is often beneficial to combine additional knowledge, if provided by the method used to perform the enrichment test; this might include the number of genes annotated to each pathway and the related subset detected as differentially expressed, but more importantly it can incorporate the full set of expression values in the original dataset.

One way to do so is via the `get_aggrscores()` function, which computes the overall pathway Z-score and an aggregated score (such as the mean or the median), which summarize at the gene set level the effect ($\log_2$FC) of the differentially expressed genes annotated as its members. In detail, the gene set Z-score attempts to determine the "direction" of change, and is computed as $z_{gs} = \frac{(DE_{up} - DE_{down})}{\sqrt{(DE_{up} + DE_{down})}}$, where DE_{up} and DE_{down} are the number of up-regulated and down-regulated DE genes annotated to the geneset *gs*, respectively (Fig. 3A, B).

Alternatively, a sample-level gene set score can be computed, in an approach similar to the implementation of `GSVA` [77]. First, a variance stabilizing transformation is applied to the expression matrix, returning values that show a higher degree of homoscedasticity, thus more amenable to downstream processing and visualization. For each gene, the values are Z-standardized by subtracting the row-wise mean and dividing by the row-wise standard deviation. Finally, for each pathway, we take the subset of the Z values corresponding to its members, and its average is computed and returned as pathway score. We define Z_{ij} as the Z-score for gene i in sample j as $Z_{ij} = \frac{T_{ij} - \bar{T}_i}{sd_i}$, whereas T_{ij} is the correspondent entry in the transformed expression values matrix, $\bar{T}_i$ and sd_i are the mean and standard deviation for the gene i, respectively. The entry GS_{kj} for pathway k in sample j is thus defined as $GS_{kj} = \frac{1}{|P_k|} \sum_{i \in P_k} Z_{ij}$, where P_k is the set of DE genes for pathway k (Fig. 3C).

This extra information about the status of activation/repression for each pathway can be efficiently encoded as aesthetic elements in plots (e.g. the color of a node in a graph, with the geneset Z-score), or directly displayed as a heatmap of the pathway score matrix to compare the activity among the samples.

Exploring the interplay of pathways and genes, interactively

The relationships among pathways and their member genes, or just between different pathways, can quickly become hard to manage when using simple textual or tabular formats. This can be due to the growing size of existing annotations, whereas the increase in detail can also lead to an increase in redundancy, thus making the task of extracting the key biological messages harder.

A number of visualization techniques have been adopted in the last years to simplify this basic yet essential operation [18], and a common way to represent this complex interplay is by using graphs. Unipartite graphs are an efficient way to depict the degree of similarity among genesets, where genesets themselves are the nodes, and edges encode for information such as the degree of similarity/overlap between the two nodes [78]—see Additional file 4: Fig. S1. Bipartite graphs (as in Fig. 2) can be naturally adopted to include both genes and genesets as the main node types, with unweighted edges representing in this case the binary membership status for one gene with respect to one geneset [79].

`GeneTonic` builds upon these foundations and implements the possibility to interact with the nodes upon hovering with the mouse (or clicking on them). The graph objects are generated dynamically, including the desired number of genesets; by default, the top most significant hits in the enrichment results are selected. Interactivity is provided by the `visNetwork` package [80], that wraps the vis.js library bindings, building on the `htmlwidgets` framework. Depending on the type of node selected in the main user interface, an information box is populated (Fig. 2D, E).

If selecting a pathway (displayed as a yellow box), the info box will contain details on the geneset (if detected as a Gene Ontology term), and a signature heatmap is displayed, with the variance stabilizing transformed expression data encoded as color to give a fine-grained view of the behavior for all its set members; this is particularly useful to connect the existing biological background with lists of features where no information on the topology is provided, enabling to detect subgroups of correlated expression patterns. Another useful representation can be obtained by coupling a volcano plot (for representing differential expression) with the annotated labels of the members of a geneset; this is implemented in the `signature_volcano()` function, and displayed in the same info box (Additional file 4: Fig. S1).

Genes are displayed in the graph as ellipses, colored using a divergent palette to encode for the effect size as $\log_2$ fold change; when a gene is selected, a plot for the corresponding expression values is shown, split by experimental variables, and the DE results for the selected gene, together with automatically generated links to external databases opening up in new tabs, to simplify the subsequent exploration steps.

The content available in the *Gene-Geneset* tab is an excellent starting point to get an overview on the provided data. While navigating the interactive graph, it might occur that the user encounters genes or genesets of particular interest; by simply clicking on the Bookmark button in the header section (or alternatively, pressing the left control key) while the node is selected, these elements are stored throughout the session and collected in the *Bookmarks* panel, where one can generate a dedicated report on these entities.

`GeneTonic` enables the extraction of a graph backbone, wrapping the efficient implementation of the `backbone` package [81] to highlight the salient edges of the bipartite projections for each type of features included, as a way to summarize information contained in large networks (Fig. 2F, G).

Additional insight can be extracted by drilling down the interactive *Enrichment Map* [62, 78], either by focusing on the selected nodes (checking out signature heatmaps or bookmarking the genesets for inserting them into the report), or also by running a variety of community detection algorithms on the graph object returned by the `enrichment_map()` function (Additional file 4: Fig. S1C). Together with the community membership information, it is then possible to obtain a more compact summary for the functional enrichment results, where the most representative genesets for each subpartition of the graph are selected and returned in tabular format. This network-based approach can be exploited to detect the handful of overarching themes, which might give a more immediate snapshot than the many, often redundant, categories, commonly returned by pathway enrichment algorithms (Additional file 4: Fig. S1E-F-G).

Summarizing the enrichment results

`GeneTonic` provides numerous ways to summarize the enrichment results, often leveraging the effectiveness of visual representations to extract insights. The *Overview* and *GSViz* panels serve this purpose, showcasing different views on the dataset at hand, with the main controls provided in the right sidebar.

The geneset volcano plot (Fig. 3A) displays all genesets from the `res_enrich` object and labels the most relevant (or any subset of interest). We use one of the aggregated scores (geneset Z-score, or average $\log_2$ fold change) to determine the horizontal position in the plot. To avoid clutter, it is also possible to reduce the terms based on an overlap threshold, retaining only the most representative ones, and provide this more compact summary to the following visualization routines.

The enhanced table (Fig. 3B) summarizes the top genesets by displaying the $\log_2$FC of each set's components along a line (one for each set). On top of the static version, this is provided also as an interactive widget, where tooltips activated with the mouse deliver extra information on each dot, representing a single gene.

The complex relationships among genesets and their behavior across samples are just two aspects one can inspect in depth with the implemented methods. Among these, users can generate a genesets-by-sample heatmap, showing the standardized expression values of the members (via the `gs_scoresheat()` function, Fig. 3C), or alternatively a summary heatmap (with `gs_summary_heat()`, Additional file 3), which aims to display the redundancy between different sets, while encoding the values of the expression changes. A multi-dimensional scaling (MDS) plot (Fig. 3D) delivers a 2d visualization of the distance among genesets, based on a similarity measure, e.g. their overlap or other criteria, such as their semantic similarity. In a similar fashion, a dendrogram for genesets enables the possibility to use node color, node size, and branch color to encode relevant features, with the tree structure mirroring the distance matrix based on a similarity measure. `GeneTonic` simplifies the creation of simple summaries for the enrichment, where the essential columns are encoded as graphical parameters of the points, extendable to the case of comparing the same genesets in more than one scenario (e.g. if it is possible to extract more than one contrast from the expression matrix). Switching to polar coordinates, this can be captured in spider plots for one or more `res_enrich` objects (see Additional file 3 for more examples of usage).

These visual summaries constitute appealing alternatives to the commonly reported tabular formats, which often fail to provide an overall view for the affected functional landscape.

Wrapping up the session

The *Bookmarks* panel offers the possibility to review and inspect the shortlisted features of interest, where both genes (on the left side of the interface) and genesets (right side) can be exported to text files.

A more comprehensive report, with dynamically generated content based on the user selections, is compiled when starting the `happy_hour()` function. This is made possible by a template RMarkdown document, included in the `GeneTonic` package, which accesses the input elements and the reactive values for the Shiny components. Notably, this functionality can also be used outside an interactive usage session, specifying as parameters the values for the genes and genesets to focus on. In either case, a full HTML document is rendered, whose content mirrors the structure of the info boxes, and can be later shared or stored as a reproducible artifact for the performed analyses.

A set of buttons below each generated visualization displays in a modal window the code necessary to reproduce a specific output—these snippets can be copied from the in-app editor into more comprehensive analysis scripts or notebooks.

Another action button creates the serialized version of a `SummarizedExperiment` object, ready to be provided as the main input to `iSEE` [74], for further tailored visualizations, either with standard or custom panels of the web application.

Discussion

Interpreting the results of transcriptomic studies can be a complex task, where differential expression analysis is combined with a higher-level pathway enrichment analysis, in order to robustly define the molecular actors that display expression changes, and also to identify the underlying functional patterns. Geneset functional enrichment has been successfully applied to thousands of works, and for this step many methods and approaches have been developed. These tasks are also often shared with alternative workflows other than DE analysis, whereas the aim is to extract meaningful information from large lists of genes, yet it is still a prohibitive task to combine in a straightforward way all the single results from each step. This can be for example due to disjoint sets of identifiers, different output and file formats, and to the difficulties in extracting knowledge while handling large numbers of redundant genesets. Providing concise and biologically meaningful views of the underlying cellular processes, defined via differential expression, is essential in many applications, and a proper visualization framework plays a fundamental role in transforming the otherwise tedious and error/bias-prone task of navigating large textual tables into a more compelling activity [18, 78].

In this work, we introduced `GeneTonic` as a solution to explore all the components of a transcriptome dataset in a more integrative way, instead of having to process them as separated outputs. As such, `GeneTonic` is not structured as an end-to-end workflow including quantification, preprocessing, exploratory data analysis, and DE modeling—all operations that are also time consuming, but in many scenarios need to be carried out only once. `GeneTonic` is focused on the analytic step devoted to the interpretation of data, rather than on the implementation of additional methods for detection of functionally enriched biological processes or pathways. Consequently, `GeneTonic` implements a variety of summary and visual representations, while accommodating the output of many commonly adopted enrichment tools, making efficient use of the Shiny framework to deliver interactivity and enable drill-down operations. These would otherwise need to be laboriously addressed in multiple iterations of scripted analyses, either done by the user itself or in collaboration with an external unit, such as a bioinformatics core facility. This approach liberates and empowers both experienced analysts and bench scientists, providing a common ground to efficiently transform data and results into interpretable discoveries.

Several software packages and web-based portals exist for providing similar functionality, and a comprehensive overview of their salient features is presented in Additional file 1: Table S1. Naturally, these tools differ in terms of implementation, range of applicability, ease of use, with many proposals offering embedded versions of enrichment tests. Since we developed `GeneTonic` in the R programming language, where many such testing procedures are natively available, we instead focused on the support and integration of their output formats into a common workflow. This can be easily combined with existing analysis pipelines, making our tool well suit for potential wide adoption. The comparison with other tools is also available online (https://federicomarini.github.io/GeneTonic_supplement), linked to a Google Sheet where the individual characteristics of each tool can be updated, in order to provide guidance for users who might be seeking advice on which solution best fits their needs (accessible at https://docs.google.com/spreadsheets/d/167XV0w18P0FSld1dt6owN4C2Esxl5FU2QTo4D-wclz0/edit?usp=sharing).

While currently focused on the output of single ORA and FCS enrichment methods, future developments of `GeneTonic` will implement functionality for combined and ensemble approaches, such as `EnrichmentBrowser` [82] or `EGSEA` [83]. Moreover, extending such visualizations and interactive summaries to scenarios where multiple omics layers are collected will be a promising avenue for `GeneTonic`, given the growing number of such datasets becoming available. Finally, we intend to address more refined similarity measurements among genesets, e.g. accounting for information contained in protein-protein interaction networks databases [84], in order to better capture the functional relatedness of the affected pathways.

As bioinformatics evolves constantly into a highly interdisciplinary field, it will become increasingly important to develop common platforms usable by many profiles with substantial differences in their level of programming skills, and `GeneTonic`'s design guidelines adhere to this principle. Serving as a bridge between experts from different disciplines, applications such as `GeneTonic` will have a significant didactic effect by making comprehensive analyses more open, transparent, and easy to share—especially in the steps of interpreting transcriptome datasets, unlocking their full potential for medical and biological research. Moreover, `GeneTonic` could make it easier for bioinformatics skilled users to better understand the systems under investigation, prompting e.g. the development of further tailored methods, which could be a key in obtaining a deeper knowledge of the experimental scenarios.

Conclusion

The identification of relevant functional patterns for the features identified in the differential expression analysis, accounting for the available expression data, remains one of the common bottlenecks for transcriptome-based workflows. `GeneTonic` provides a web application and many underlying functions to assemble the pieces together, supporting the exploration both interactively as well as in a programmatic way. Combining together the results for quantification, DE testing, and functional enrichment (either generated autonomously, or obtained from collaborators), `GeneTonic` assists in the unmet yet increasing need of extracting novel knowledge and insights, which can become daunting especially on larger datasets.

`GeneTonic` has the potential to become an ideal interface between experimental and computational scientists, with the HTML report built via RMarkdown as a milestone for reproducibility, upon conclusion of an interactive session. `GeneTonic` can be integrated in a wide spectrum of existing bioinformatic pipelines, as it provides functions to convert and input the results of many pathway enrichment tools. This aligns with the principle of interoperability at the heart of the Bioconductor project, which enables a large number of such workflows.

The experience of enjoying transcriptomic data analysis and exploration can be easily shared with reduced communication burden, with both experimental and computational sides empowered in the tasks of realizing complex summaries and visualizations. This will significantly facilitate and democratize the discovery process, bridging the gaps existing between technical and domain expertise.

Availability and requirements

Project name: GeneTonic.

Project home page: https://bioconductor.org/packages/GeneTonic/ (release), https://github.com/federicomarini/GeneTonic/ (development version).

Archived version: https://doi.org/10.5281/zenodo.5519859, package source as gzipped tar archive of the version reported in this article

Project documentation: rendered at https://federicomarini.github.io/GeneTonic/.

Operating systems: Linux, Mac OS, Windows.

Programming language: R.

Other requirements: R-4.0.0 or higher, Bioconductor 3.11 or higher.

License: MIT.

Any restrictions to use by non-academics: none.

Abbreviations

DE: Differential expression; FCS: Functional class scoring; FDR: False discovery rate; GO: Gene ontology; GSEA: Gene set enrichment analysis; HGNC: HUGO (Human Genome Organisation) Gene Nomenclature Committee; $\log_2$FC: Base-2 logarithm of the fold change; MA plot: M (log ratio) versus A (mean average) plot; MDS: Multi-dimensional scaling; MSigDB: Molecular Signatures Database; NCBI: National Center for Biotechnology Information; ORA: Over-representation analysis; PT: Pathway topology; RNA-seq: RNA sequencing.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04461-5.

Additional file 1: Table S1. Comparison of tools for assisting the interpretation of transcriptome data via functional enrichment analyses, including link to the related publications and to the source code repositories (if available). Evaluation criteria are included in the dedicated sheet. The information contained in this table are also available online at https://federicomarini.github.io/GeneTonic_supplement, displaying the content of the Google Sheet, accessible at https://docs.google.com/spreadsheets/d/1EQD8gNk6NID_8CJ-xetQm31XB4eRCXIsvVd4X0oaoPo/edit?usp=sharing).

Additional file 2. Complete use case for the `GeneTonic` package, based on the RNA-seq dataset for macrophage immune stimulation (Interferon Gamma treatment vs naive cells, ERP020977).

Additional file 3. Complete use case for the `GeneTonic` package on the RNA-seq A20-deficient microglia dataset (A20 deficiency vs control, GSE123033).

Additional file 4: Fig. S1. Screenshot of the *Enrichment Map* panel in the `GeneTonic` application. The sidebar menu (**A**) controls the main navigation in the app, and a common set of options is toggled with the cogs icon (**B**). The main area of the *Enrichment Map* panel (**C**) contains an interactive graph for the enrichment map of the genesets, connected according to their similarity, and color coded according to the specified geneset property (here, the Z-score). Upon clicking on any geneset, a Geneset Box (**D**) is displayed for further exploration (e.g. to show a volcano plot with the geneset members labelled). The geneset distillery (**E**) enables the exploration of meta-genesets, derived by computing clusters on the graph object underlying the enrichment map. From the tabular representation, it is possible to visualize meta-genesets as heatmaps (**F**), or display a modal popup containing the enrichment map where the cluster assignments of the genesets are shown (**G**).

Acknowledgements

This work has been supported by the computing infrastructure provided by the Core Facility Bioinformatics at the University Medical Center Mainz, used also for deploying the demo instance. The authors thank the members of the Core Facility Bioinformatics at the Institute of Molecular Biology Mainz, Miguel Andrade (IOME, Johannes Gutenberg University of Mainz), Gerrit Toenges and Arsenij Ustjanzew (IMBEI Mainz), and Francesca Finotello (ICBI, Medical University of Innsbruck) for valuable feedback and suggestions.

Authors' contributions

FM—conceptualization, data curation, formal analysis, funding acquisition, methodology, project administration, resources, software, supervision, visualization, writing—original draft, writing—review and editing. AL—data curation, formal analysis, methodology, software, visualization, writing—review and editing. JL—data curation, software, writing—review and editing. KS—conceptualization, funding acquisition, resources, supervision, writing—original draft, writing—review and editing. All authors read and approved the final version of the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. The work of FM is supported by the German Federal Ministry of Education and Research (BMBF 01EO1003). The funding body had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Availability of data and materials

The datasets used in this manuscript and its supplements are available from the following articles: The data set on the macrophage immune stimulation is included in PubMed ID: 29379200 (https://doi.org/10.1038/s41588-018-0046-7). Dataset deposited at the ENA (ERP020977, project id: PRJEB18997) and accessed from the Bioconductor experiment package `macrophage` package (https://bioconductor.org/packages/macrophage/, version 1.7.2). The data set on murine A20-deficient microglia is included in PubMed ID: 32023471 (https://doi.org/10.1016/j.celrep.2019.12.097).

Dataset deposited at the GEO (GSE123033, project id: PRJNA507355) and accessed from the https://github.com/federicomarini/GeneTonic_supplement/ repository. The `GeneTonic` package can be downloaded from its Bioconductor page https://bioconductor.org/packages/GeneTonic/ or the GitHub development page https://github.com/federicomarini/GeneTonic/. `GeneTonic` is also provided as a recipe in Bioconda (https://anaconda.org/bioconda/bioconductor-genetonic). The repository available at https://github.com/federicomarini/GeneTonic_supplement/ contains the code used to generate the supplemental material, and the required input data to replicate the analyses presented in the use cases.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg University Mainz, Obere Zahlbacher Str. 69, 55131 Mainz, Germany. [2]Center for Thrombosis and Hemostasis (CTH), University Medical Center of the Johannes Gutenberg University Mainz, Langenbeckstr. 1, 55131 Mainz, Germany.

Received: 25 May 2021 Accepted: 26 October 2021
Published online: 23 December 2021

References

1. Van den Berge K, Hembach KM, Soneson C, Tiberi S, Clement L, Love MI, Patro R, Robinson MD. RNA sequencing data: Hitchhikers guide to expression analysis. Annu Rev Biomed Data Sci. 2019;2(1):139–73. https://doi.org/10.1146/annurev-biodatasci-072018-021255.
2. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, Szcześniak MW, Gaffney DJ, Elo LL, Zhang X, Mortazavi A. A survey of best practices for RNA-seq data analysis. Genome Biol. 2016;17(1):13. https://doi.org/10.1186/s13059-016-0881-8.
3. Love MI, Anders S, Kim V, Huber W. RNA-Seq workflow: gene-level exploratory analysis and differential expression. F1000Research. 2015;4:1070. https://doi.org/10.12688/f1000research.7035.1.
4. Chen Y, Lun ATL, Smyth GK. From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline. F1000Research. 2016;5:1438. https://doi.org/10.12688/f1000research.8987.2.
5. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. Nat Genet. 2000;25(1):25–9. https://doi.org/10.1038/75556.
6. Carbon S, Douglass E, Dunn N, Good B, Harris NL, Lewis SE, Mungall CJ, Basu S, Chisholm RL, Dodson RJ, Hartline E, Fey P, Thomas PD, Albou LP, Ebert D, Kesling MJ, Mi H, Muruganujan A, Huang X, Poudel S, Mushayahama T, Hu JC, LaBonte SA, Siegele DA, Antonazzo G, Attrill H, Brown NH, Fexova S, Garapati P, Jones TEM, Marygold SJ, Millburn GH, Rey AJ, Trovisco V, Dos Santos G, Emmert DB, Falls K, Zhou P, Goodman JL, Strelets VB, Thurmond J, Courtot M, Osumi DS, Parkinson H, Roncaglia P, Acencio ML, Kuiper M, Lreid A, Logie C, Lovering RC, Huntley RP, Denny P, Campbell NH, Kramarz B, Acquaah V, Ahmad SH, Chen H, Rawson JH, Chibucos MC, Giglio M, Nadendla S, Tauber R, Duesbury MJ, Del NT, Meldal BHM, Perfetto L, Porras P, Orchard S, Shrivastava A, Xie Z, Chang HY, Finn RD, Mitchell AL, Rawlings ND, Richardson L, Sangrador-Vegas A, Blake JA, Christie KR, Dolan ME, Drabkin HJ, Hill DP, Ni L, Sitnikov D, Harris MA, Oliver SG, Rutherford K, Wood V, Hayles J, Bahler J, Lock A, Bolton ER, De Pons J, Dwinell M, Hayman GT, Laulederkind SJF, Shimoyama M, Tutaj M, Wang SJ, D'Eustachio P, Matthews L, Balhoff JP, Aleksander SA, Binkley G, Dunn BL, Cherry JM, Engel SR, Gondwe F, Karra K, MacPherson KA, Miyasato SR, Nash RS, Ng PC, Sheppard TK, Shrivatsav Vp A, Simison M, Skrzypek MS, Weng S, Wong ED, Feuermann M, Gaudet P, Bakker E, Berardini TZ, Reiser L, Subramaniam S, Huala E, Arighi C, Auchincloss A, Axelsen K, Argoud GP, Bateman A, Bely B, Blatter MC, Boutet E, Breuza L, Bridge A, Britto R, Bye-A-Jee H, Casals-Casas C, Coudert E, Estreicher A, Famiglietti L, Garmiri P, Georghiou G, Gos A, Gruaz-Gumowski N, Hatton-Ellis E, Hinz U, Hulo C, Ignatchenko A, Jungo F, Keller G, Laiho K, Lemercier P, Lieberherr D, Lussi Y, Mac-Dougall A, Magrane M, Martin MJ, Masson P, Natale DA, Hyka NN, Pedruzzi I, Pichler K, Poux S, Rivoire C, Rodriguez-Lopez M, Sawford T, Speretta E, Shypitsyna A, Stutz A, Sundaram S, Tognolli M, Tyagi N, Warner K, Zaru R, Wu C, Chan J, Cho J, Gao S, Grove C, Harrison MC, Howe K, Lee R, Mendel J, Muller HM, Raciti D, Van Auken K, Berriman M, Stein L, Sternberg PW, Howe D, Toro S, Westerfield M. The gene ontology resource: 20 years and still going strong. Nucleic Acids Res. 2019;47(D1):330–8. https://doi.org/10.1093/nar/gky1055.
7. Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K. KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 2017;45(D1):353–61. https://doi.org/10.1093/nar/gkw1092.
8. Kanehisa M, Sato Y, Furumichi M, Morishima K, Tanabe M. New approach for understanding genome variations in KEGG. Nucleic Acids Res. 2019;47(D1):590–5. https://doi.org/10.1093/nar/gky962.
9. Fabregat A, Jupe S, Matthews L, Sidiropoulos K, Gillespie M, Garapati P, Haw R, Jassal B, Korninger F, May B, Milacic M, Roca CD, Rothfels K, Sevilla C, Shamovsky V, Shorser S, Varusai T, Viteri G, Weiser J, Wu G, Stein L, Hermjakob H,

D'Eustachio P. The reactome pathway knowledgebase. Nucleic Acids Res. 2018;46(D1):649–55. https://doi.org/10.1093/nar/gkx1132.

10. Liberzon A., Subramanian A., Pinchback R., Thorvaldsdottir H., Tamayo P., Mesirov J.P. Molecular signatures database (MSigDB) 3.0. Bioinformatics. 2011;27(12):1739–40. https://doi.org/10.1093/bioinformatics/btr260.
11. Liberzon A, Birger C, Thorvaldsdóttir H, Ghandi M, Mesirov JP, Tamayo P. The molecular signatures database Hallmark gene set collection. Cell Syst. 2015;1(6):417–25. https://doi.org/10.1016/j.cels.2015.12.004.
12. Khatri P, Sirota M, Butte AJ. Ten years of pathway analysis: current approaches and outstanding challenges. PLoS Comput Biol. 2012;8(2):1002375. https://doi.org/10.1371/journal.pcbi.1002375.
13. Xie C, Jauhari S, Mora A. Popularity and performance of bioinformatics software: the case of gene set analysis. BMC Bioinform. 2021;22(1):191. https://doi.org/10.1186/s12859-021-04124-5.
14. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci. 2005;102(43):15545–50. https://doi.org/10.1073/pnas.0506580102.
15. Nguyen T, Mitrea C, Draghici S. Network-based approaches for pathway level analysis. Curr Protoc Bioinform. 2018;61(1):8–25182524. https://doi.org/10.1002/cpbi.42.
16. Geistlinger L, Csaba G, Santarelli M, Ramos M, Schiffer L, Turaga N, Law C, Davis S, Carey V, Morgan M, Zimmer R, Waldron L. Toward a gold standard for benchmarking gene set enrichment analysis. Brief Bioinform. 2020. https://doi.org/10.1093/bib/bbz158.
17. Villaveces JM, Koti P, Habermann BH. Tools for visualization and analysis of molecular networks, pathways, and -omics data. Adv Appl Bioinform Chem. 2015;8(1):11–22. https://doi.org/10.2147/AABC.S63534.
18. Supek F, Škunca N, Visualizing GO annotations. In: The gene ontology handbook, vol. 1446. Humana Press; 2017. p. 207–20. https://doi.org/10.1007/978-1-4939-3743-1.
19. Maere S, Heymans K, Kuiper M. BiNGO: a Cytoscape plugin to assess over representation of gene ontology categories in biological networks. Bioinformatics. 2005;21(16):3448–9. https://doi.org/10.1093/bioinformatics/bti551.
20. Bindea G, Mlecnik B, Hackl H, Charoentong P, Tosolini M, Kirilovsky A, Fridman WH, Pagès F, Trajanoski Z, Galon J. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. Bioinformatics. 2009;25(8):1091–3. https://doi.org/10.1093/bioinformatics/btp101.
21. Mlecnik B, Galon J, Bindea G. Comprehensive functional analysis of large lists of genes and proteins. J Proteomics. 2018;171:2–10. https://doi.org/10.1016/j.jprot.2017.03.016.
22. Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. BMC Bioinform. 2009;10(1):48. https://doi.org/10.1186/1471-2105-10-48.
23. Supek F, Bošnjak M, Škunca N, Šmuc T. REVIGO summarizes and visualizes long lists of gene ontology terms. PLoS ONE. 2011;6(7):21800. https://doi.org/10.1371/journal.pone.0021800.
24. Walter W, Sánchez-Cabo F, Ricote M. GOplot: an R package for visually combining expression data with functional analysis. Bioinformatics. 2015;31(17):2912–4. https://doi.org/10.1093/bioinformatics/btv300.
25. Tian T, Liu Y., Yan H, You Q., Yi X., Du Z., Xu W., Su Z. AgriGO v2.0: a GO analysis toolkit for the agricultural community, 2017 update. Nucleic Acids Res. 2017;45(W1):122–9. https://doi.org/10.1093/nar/gkx382.
26. Wei Q, Khan IK, Ding Z, Yerneni S, Kihara D. NaviGO: interactive tool for visualization and functional similarity and coherence analysis with gene ontology. BMC Bioinform. 2017;18(1):177. https://doi.org/10.1186/s12859-017-1600-5.
27. Liao Y, Wang J, Jaehnig EJ, Shi Z, Zhang B. WebGestalt 2019: gene set analysis toolkit with revamped UIs and APIs. Nucleic Acids Res. 2019;47(W1):199–205. https://doi.org/10.1093/nar/gkz401.
28. Kuznetsova I, Lugmayr A, Siira SJ, Rackham O, Filipovska A. CirGO: an alternative circular way of visualising gene ontology terms. BMC Bioinform. 2019;20(1):84. https://doi.org/10.1186/s12859-019-2671-2.
29. Zhu J, Zhao Q, Katsevich E, Sabatti C. Exploratory gene ontology analysis with interactive visualization. Sci Rep. 2019;9(1):1–9. https://doi.org/10.1038/s41598-019-42178-x.
30. Hale ML, Thapa I, Ghersi D. FunSet: an open-source software and web server for performing and displaying gene ontology enrichment analysis. BMC Bioinform. 2019;20(1):359. https://doi.org/10.1186/s12859-019-2960-9.
31. Federico A, Monti S. hypeR: an R package for geneset enrichment workflows. Bioinformatics. 2020;36(4):1307–8. https://doi.org/10.1093/bioinformatics/btz700.
32. Liu X, Han M, Zhao C, Chang C, Zhu Y, Ge C, Yin R, Zhan Y, Li C, Yu M, He F, Yang X. KeggExp: a web server for visual integration of KEGG pathways and expression profile data. Bioinformatics. 2019;35(8):1430–2. https://doi.org/10.1093/bioinformatics/bty798.
33. Zhou Y, Zhou B, Pache L, Chang M, Khodabakhshi AH, Tanaseichuk O, Benner C, Chanda SK. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. Nat Commun. 2019;10(1):1523. https://doi.org/10.1038/s41467-019-09234-6.
34. Ulgen E, Ozisik O, Sezerman O.U. pathfindR: an R package for comprehensive identification of enriched pathways in omics data through active subnetworks. Front Genet. 2019;10(SEP):1–33. https://doi.org/10.3389/fgene.2019.00858.
35. Ge SX, Jung D, Yao R. ShinyGO: a graphical gene-set enrichment tool for animals and plants. Bioinformatics. 2020;36(8):2628–9. https://doi.org/10.1093/bioinformatics/btz931.
36. Brionne A, Juanchich A, Hennequet-Antier C. ViSEAGO: a bioconductor package for clustering biological functions using gene ontology and semantic similarity. BioData Min. 2019;12(1):1–13. https://doi.org/10.1186/s13040-019-0204-1.
37. Szklarczyk D, Gable AL, Lyon D, Junge A, Wyder S, Huerta-Cepas J, Simonovic M, Doncheva NT, Morris JH, Bork P, Jensen LJ, von Mering C. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res. 2019;47(D1):607–13. https://doi.org/10.1093/nar/gky1131.
38. Tokar T, Pastrello C, Jurisica I. GSOAP: a tool for visualisation of gene set over-representation analysis. Bioinformatics. 2020. https://doi.org/10.1093/bioinformatics/btaa001.
39. Wang G, Oh D-H, Dassanayake M. GOMCL: a toolkit to cluster, evaluate, and extract non-redundant associations of gene ontology-based functions. BMC Bioinform. 2020;21(1):139. https://doi.org/10.1186/s12859-020-3447-4.

40. Kim J, Yoon S, Nam D. netGO: R-Shiny package for network-integrated pathway enrichment analysis. Bioinformatics. 2020. https://doi.org/10.1093/bioinformatics/btaa077.
41. Calura E, Martini P. Summarizing RNA-Seq data or differentially expressed genes using gene set, network, or pathway analysis. In: Picardi E, editor. RNA bioinformatics, chap 9, vol. 2284. Humana; 2021. p. 147–79. https://doi.org/10.1007/978-1-0716-1307-8.
42. Akhmedov M, Martinelli A, Geiger R, Kwee I. Omics Playground: a comprehensive self-service platform for visualization, analytics and exploration of big omics data. NAR Genom Bioinform. 2020;2(1):1–10. https://doi.org/10.1093/nargab/lqz019.
43. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten simple rules for reproducible computational research. PLoS Comput Biol. 2013;9(10):1003285. https://doi.org/10.1371/journal.pcbi.1003285.
44. Marini F, Binder H. Development of applications for interactive and reproducible research: a case study. Genom Computl Biol. 2016;3(1):39. https://doi.org/10.18547/gcb.2017.vol3.iss1.e39.
45. Brito JJ, Li J, Moore JH, Greene CS, Nogoy NA, Garmire LX, Mangul S. Recommendations to enhance rigor and reproducibility in biomedical research. GigaScience. 2020;9(6):1–6. https://doi.org/10.1093/gigascience/giaa056.
46. Knuth DE. Literate programming. Comput J. 1984;27(2):97–111. https://doi.org/10.1093/comjnl/27.2.97.
47. Marini F, Binder H. pcaExplorer: an R/Bioconductor package for interacting with RNA-seq principal components. BMC Bioinform. 2019;20(1):331. https://doi.org/10.1186/s12859-019-2879-1.
48. Marini F, Linke J, Binder H. ideal: an R/Bioconductor package for interactive differential expression analysis. BMC Bioinform. 2020;21(1):565. https://doi.org/10.1186/s12859-020-03819-5.
49. Poplawski A, Marini F, Hess M, Zeller T, Mazur J, Binder H. Systematically evaluating interfaces for RNA-seq analysis from a life scientist perspective. Brief Bioinform. 2016;17(2):213–23. https://doi.org/10.1093/bib/bbv036.
50. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, Bravo HC, Davis S, Gatto L, Girke T, Gottardo R, Hahne F, Hansen KD, Irizarry R, Lawrence M, Love MI, MacDonald J, Obenchain V, Oleś AK, Pagès H, Reyes A, Shannon P, Smyth GK, Tenenbaum D, Waldron L, Morgan M. Orchestrating high-throughput genomic analysis with bioconductor. Nat Methods. 2015;12(2):115–21. https://doi.org/10.1038/nmeth.3252.
51. Amezquita R, Carey V, Carpp L, Geistlinger L, Lun A, Marini F, Rue-Albrecht K, Risso D, Soneson C, Waldron L, Pagès H, Smith M, Huber W, Morgan M, Gottardo R, Hicks S. Orchestrating single-cell analysis with bioconductor. BioRxiv. 2019. https://doi.org/10.1101/590562.
52. Chang W, Cheng J, Allaire J, Xie Y, McPherson J. Shiny: web application framework for R. (2020). R package version 1.4.0.2. https://CRAN.R-project.org/package=shiny.
53. Alasoo K, Rodrigues J, Mukhopadhyay S, Knights AJ, Mann AL, Kundu K, Hale C, Dougan G, Gaffney DJ. Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response. Nat Genet. 2018;50(3):424–31. https://doi.org/10.1038/s41588-018-0046-7.
54. Mohebiany AN, Ramphal NS, Karram K, Di Liberto G, Novkovic T, Klein M, Marini F, Kreutzfeldt M, Härtner F, Lacher SM, Bopp T, Mittmann T, Merkler D, Waisman A. Microglial A20 protects the brain from CD8 T-cell-mediated immunopathology. Cell Rep. 2020;30(5):1585–15976. https://doi.org/10.1016/j.celrep.2019.12.097.
55. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014;15(12):550. https://doi.org/10.1186/s13059-014-0550-8.
56. Yates AD, Achuthan P, Akanni W, Allen J, Allen J, Alvarez-Jarreta J, Amode MR, Armean IM, Azov AG, Bennett R, Bhai J, Billis K, Boddu S, Marugán JC, Cummins C, Davidson C, Dodiya K, Fatima R, Gall A, Giron CG, Gil L, Grego T, Haggerty L, Haskell E, Hourlier T, Izuogu OG, Janacek SH, Juettemann T, Kay M, Lavidas I, Le T, Lemos D, Martinez JG, Maurel T, McDowall M, McMahon A, Mohanan S, Moore B, Nuhn M, Oheh DN, Parker A, Parton A, Patricio M, Sakthivel MP, Abdul Salam AI, Schmitt BM, Schuilenburg H, Sheppard D, Sycheva M, Szuba M, Taylor K, Thormann A, Threadgold G, Vullo A, Walts B, Winterbottom A, Zadissa A, Chakiachvili M, Flint B, Frankish A, Hunt SE, IIsley G, Kostadima M, Langridge N, Loveland JE, Martin FJ, Morales J, Mudge JM, Muffato M, Perry E, Ruffier M, Trevanion SJ, Cunningham F, Howe KL, Zerbino DR, Flicek P. Ensembl 2020. Nucleic Acids Res. 2019;48(D1):682–8. https://doi.org/10.1093/nar/gkz966.
57. Frankish A, Diekhans M, Ferreira AM, Johnson R, Jungreis I, Loveland J, Mudge JM, Sisu C, Wright J, Armstrong J, Barnes I, Berry A, Bignell A, Carbonell Sala S, Chrast J, Cunningham F, Di Domenico T, Donaldson S, Fiddes IT, García Girón C, Gonzalez JM, Grego T, Hardy M, Hourlier T, Hunt T, Izuogu OG, Lagarde J, Martin FJ, Martínez L, Mohanan S, Muir P, Navarro FCP, Parker A, Pei B, Pozo F, Ruffier M, Schmitt BM, Stapleton E, Suner MM, Sycheva I, Uszczynska-Ratajczak B, Xu J, Yates A, Zerbino D, Zhang Y, Aken B, Choudhary JS, Gerstein M, Guigó R, Hubbard TJP, Kellis M, Paten B, Reymond A, Tress ML, Flicek P. GENCODE reference annotation for the human and mouse genomes. Nucleic Acids Res. 2019;47(D1):766–73. https://doi.org/10.1093/nar/gky955.
58. Granjon D. bs4Dash: a 'Bootstrap 4' Version of 'shinydashboard'. 2019. https://rinterface.github.io/bs4Dash/index.html, https://github.com/RinteRface/bs4Dash.
59. Chang W, Borges Ribeiro B. Shinydashboard: create dashboards with 'Shiny'. (2018). R package version 0.7.1. https://CRAN.R-project.org/package=shinydashboard.
60. Ganz C. rintrojs: a wrapper for the intro. js library. J Open Source Softw. 2016;1(6):2016. https://doi.org/10.21105/joss.00063.
61. Alexa A, Rahnenführer J, Lengauer T. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. Bioinformatics. 2006;22(13):1600–7. https://doi.org/10.1093/bioinformatics/btl140.
62. Yu G, Wang L-G, Han Y, He Q-Y. clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS J Integr Biol. 2012;16(5):284–7. https://doi.org/10.1089/omi.2011.0118.
63. Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc. 2009;4(1):44–57. https://doi.org/10.1038/nprot.2008.211.
64. Kuleshov MV, Jones MR, Rouillard AD, Fernandez NF, Duan Q, Wang Z, Koplev S, Jenkins SL, Jagodnik KM, Lachmann A, McDermott MG, Monteiro CD, Gundersen GW, Ma'ayan A. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res. 2016;44(W1):90–7. https://doi.org/10.1093/nar/gkw377.

65. Reimand J, Isserlin R, Voisin V, Kucera M, Tannus-Lopes C, Rostamianfar A, Wadi L, Meyer M, Wong J, Xu C, Merico D, Bader GD. Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. Nat Protoc. 2019;14(2):482–517. https://doi.org/10.1038/s41596-018-0103-9.
66. Raudvere U, Kolberg L, Kuzmin I, Arak T, Adler P, Peterson H, Vilo J. g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). Nucleic Acids Res. 2019;47(W1):191–8. https://doi.org/10.1093/nar/gkz369.
67. Korotkevich G, Sukhov V, Budin N, Shpak B, Artyomov MN, Sergushichev A. Fast gene set enrichment analysis. bioRxiv. 2021. https://doi.org/10.1101/060012.
68. Agarwala R, Barrett T, Beck J, Benson DA, Bollin C, Bolton E, Bourexis D, Brister JR, Bryant SH, Canese K, Charowhas C, Clark K, DiCuccio M, Dondoshansky I, Feolo M, Funk K, Geer LY, Gorelenkov V, Hlavina W, Hoeppner M, Holmes B, Johnson M, Khotomlianski V, Kimchi A, Kimelman M, Kitts P, Klimke W, Krasnov S, Kuznetsov A, Landrum MJ, Landsman D, Lee JM, Lipman DJ, Lu Z, Madden TL, Madej T, Marchler-Bauer A, Karsch-Mizrachi I, Murphy T, Orris R, Ostell J, O'Sullivan C, Palanigobu V, Panchenko AR, Phan L, Pruitt KD, Rodarmer K, Rubinstein W, Sayers EW, Schneider V, Schoch CL, Schuler GD, Sherry ST, Sirotkin K, Siyan K, Slotta D, Soboleva A, Soussov V, Starchenko G, Tatusova TA, Todorov K, Trawick BW, Vakatov D, Wang Y, Ward M, Wilbur WJ, Yaschenko E, Zbicz K. Database resources of the national center for biotechnology information. Nucleic Acids Res. 2017;45(D1):12–7. https://doi.org/10.1093/nar/gkw1071.
69. Stelzer G, Rosen N, Plaschkes I, Zimmerman S, Twik M, Fishilevich S, Stein TI, Nudel R, Lieder I, Mazor Y, Kaplan S, Dahary D, Warshawsky D, Guan-Golan Y, Kohn A, Rappaport N, Safran M, Lancet D. The GeneCards suite: from gene data mining to disease genome sequence analyses. Curr Protoc Bioinform. 2016;54(1):1–30113033. https://doi.org/10.1002/cpbi.5.
70. Gamazon ER, Segrè AV, van de Bunt M, Wen X, Xi HS, Hormozdiari F, Ongen H, Konkashbaev A, Derks EM, Aguet F, Quan J, Nicolae DL, Eskin E, Kellis M, Getz G, McCarthy MI, Dermitzakis ET, Cox NJ, Ardlie KG. Using an atlas of gene regulation across 44 human tissues to inform complex disease- and trait-associated variation. Nat Genet. 2018;50(7):956–67. https://doi.org/10.1038/s41588-018-0154-4.
71. Xie Y. Dynamic Documents with R and Knitr, p. 188. Chapman & Hall/CRC; 2013. https://doi.org/10.18637/jss.v056.b02. arXiv:arXiv:1501.0228. http://www.crcpress.com/product/isbn/9781482203530.
72. Rule A, Birmingham A, Zuniga C, Altintas I, Huang SC, Knight R, Moshiri N, Nguyen MH, Rosenthal SB, Pérez F, Rose PW. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLoS Comput Biol. 2019;15(7):1–8. https://doi.org/10.1371/journal.pcbi.1007007.
73. Stodden V, Miguez S. Best practices for computational science: software infrastructure and environments for reproducible and extensible research. J Open Res Softw. 2014;2(1):21. https://doi.org/10.5334/jors.ay.
74. Rue-Albrecht K, Marini F, Soneson C, Lun ATL. iSEE: interactive summarized experiment explorer. F1000Research. 2018;7:741. https://doi.org/10.12688/f1000research.14966.1.
75. Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C, Salmon provides fast and bias-aware quantification of transcript expression. Nat Methods. 2017. https://doi.org/10.1038/nmeth.4197. arXiv:1505.02710.
76. Lun ATL, Chen Y, Smyth GK. It's DE-licious: a recipe for differential expression analyses of RNA-seq experiments using quasi-likelihood methods in edgeR. In: Mathé E, Davis S, editors. Statistical genomics, chap. 19. Humana Press; 2016. p. 391–416.
77. Hänzelmann S, Castelo R, Guinney J. GSVA: gene set variation analysis for microarray and RNA-Seq data. BMC Bioinform. 2013;14:12. https://doi.org/10.1186/1471-2105-14-7.
78. Merico D, Isserlin R, Stueker O, Emili A, Bader GD. Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. PLoS ONE. 2010;5(11):13984. https://doi.org/10.1371/journal.pone.0013984.
79. Pomaznoy M, Ha B, Peters B. GOnet: a tool for interactive gene ontology analysis. BMC Bioinform. 2018;19(1):1–8. https://doi.org/10.1186/s12859-018-2533-3.
80. Almende BV, Thieurmel B, Robert T. visNetwork: network visualization using 'vis.js' library. (2019). R package version 2.0.9. https://CRAN.R-project.org/package=visNetwork.
81. Domagalski R, Neal ZP, Sagan B. Backbone: an R package for extracting the backbone of bipartite projections. PLoS ONE. 2021;16(1):0244363. https://doi.org/10.1371/journal.pone.0244363.
82. Geistlinger L, Csaba G, Zimmer R. Bioconductor's EnrichmentBrowser: seamless navigation through combined results of set- & network-based enrichment analysis. BMC Bioinform. 2016;17(1):45. https://doi.org/10.1186/s12859-016-0884-1.
83. Alhamdoosh M, Ng M, Wilson NJ, Sheridan JM, Huynh H, Wilson MJ, Ritchie ME. Combining multiple tools outperforms individual methods in gene set enrichment analyses. Bioinformatics. 2016;33:623. https://doi.org/10.1093/bioinformatics/btw623.
84. Yoon S, Kim J, Kim S-K, Baik B, Chi S-M, Kim S-Y, Nam D. GScluster: network-weighted gene-set clustering analysis. BMC Genom. 2019;20(1):352. https://doi.org/10.1186/s12864-019-5738-6.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.