Enhancing Analysis and Interpretation Workflows for Transcriptome Data with an Interactive R/Bioconductor Toolkit

Dissertation
Zur Erlangung des Grades
Doktor der Naturwissenschaften

Am Fachbereich Biologie
Der Johannes Gutenberg-Universität Mainz

Annekathrin Silvia Nedwed geb. Ludt
[redacted]

Mainz, 2024

Dekan:
1. Berichterstatter:
2. Berichterstatter:
Tag der mündlichen Prüfung: 20. Dezember 2024

Declaration:

Parts of this thesis are based on work that has been published or are in preparation for publication. Notably, we have published one protocol article titled "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022], which will be discussed throughout the thesis, especially in Section 3.1. For clarification, when "we" or "our" is used in this thesis, it refers to the collective authorship of the previously published work in Ludt et al. [2022].

Furthermore, R packages developed by our group, such as `pcaExplorer` [Marini, Binder, 2019], `ideal` [Marini et al., 2020], and `GeneTonic` [Marini et al., 2021], have already been published and were used in the sections on the data analysis presented in Chapter 3, specifically in Section 3.3.2 and Section 3.3.3.

I am also in the process of preparing a publication on `GeDi`, the R/Bioconductor package presented in this thesis. Moreover, it is important to note that `GeDi` will remain under active development, meaning that while this thesis describes the state of the package at the time of writing, future updates may alter, extend, or potentially replace certain functionality described herein.

Additionally, throughout this work, function names as well as arguments of the `GeDi` package are presented in **`bold monospace`** font, while objects, parameters, functions or names of other R package are written in `monospace` font. For improved readability, function parameters will be omitted unless essential, with key parameters discussed in the text. Readers are encouraged to consult the package documentation of `GeDi` at `https://annekathrinsilvia.github.io/GeDi/reference/index.html` for more detailed information and descriptions.

Table of Contents

Abbreviations

3D	Three-Dimensional
A	Adenine
App	Application
Areg	Amphiregulin
Batf	Basic Leucine Zipper Transcription Factor, ATF-like
BM	Bone Marrow
BP	Biological Process
BPPARAM	BiocParallelParam
C	Cytosine
CC	Cellular Component
Cc2d2a	Coiled-Coil and C2 Domain Containing 2A
Ccr7	C-C Motif Chemokine Receptor 7
Ccr10	C-C Motif Chemokine Receptor 10
cDNA	Complementary Deoxyribonucleic Acid
CRAN	The Comprehensive R Archive Network
CSV	Comma-Separated Values
Ctla4	Cytotoxic T-Lymphocyte-Associated Protein 4
DAG	Directed Acyclic Graph
DAVID	Database for Annotation, Visualisation, and Integrated Discovery
DE	Differential Expression
DEGs	Differentially Expressed Genes
DNA	Deoxyribonucleic Acid
EDA	Exploratory Data Analysis
FC	Fold Change
FCS	Functional Class Scoring
FDR	False Discovery Rate
G	Guanine
GeDi	Gene Set Distances
GEO	Gene Expression Omnibus
GFP	Green Fluorescent Protein
GLM	Generalised Linear Model
GO	Gene Ontology
HGNC	HUGO Gene Nomenclature Committee
HTML	Hyper Text Markup Language
IC	Information Content
ID	Identifier
Il10	Interleukin 10
KEGG	Kyoto Encyclopedia of Genes and Genomes
Klrg1	Killer Cell Lectin-Like Receptor G1
log2FC	Logarithm to Base 2 of the Fold Change
log10	Logarithm to Base 10
MF	Molecular Function
MICA	Most Informative Common Ancestor
MM	Meet-Min (Distance Score)
MSigDB	Molecular Signatures Database
mRNA	Messenger Ribonucleic Acid

Nfil3	Nuclear Factor Interleukin 3 Regulated
ORA	Over-Representation Analysis
PAM	Partitioning Around Medoids
PC	Principal Component
PCA	Principal Component Analysis
PCR	Polymerase Chain Reaction
pMM	Protein-Protein Interaction-Weighted Meet-Min (Distance Score)
PPI	Protein-Protein Interaction
PT	Pathway Topology
RNA	Ribonucleic Acid
RNA-seq	Ribonucleic Acid Sequencing
rRNA	Ribosomal Ribonucleic Acid
scRNA-seq	Single Cell Ribonucleic Acid Sequencing
ST2	Suppression of Tumorigenicity 2
STRING	Search Tool for Recurring Instances of Neighbouring Genes
T	Thymine
Tigit	T Cell Immunoreceptor with Ig and ITIM Domains
Treg	Regulatory T (Cell)
tRNA	Transfer Ribonucleic Acid
U	Uracil
UI	User Interface

Abstract

Over the recent years, bulk RNA-sequencing (RNA-seq) has become the gold standard for transcriptome analysis, leading to a significant increase in the volume of data and results being generated. Consequently, this growth has also made the task of interpreting these results increasingly challenging, particularly for functional enrichment analyses.

Functional enrichment analysis constitutes a fundamental step in the analysis of various omics datasets, aiming to identify differentially regulated pathways between experimental conditions and to draw insights into the underlying molecular mechanisms of diseases and specific phenotypes. Due to this widespread use, there are numerous tools and implementations available to calculate these results. Despite their utility, existing methods often yield impractical outputs comprising extensive lists of gene sets, impeding hypothesis generation and synthesis due to inherent redundancy in the found pathways. Additionally, prevalent approaches for processing enrichment results lack consideration of network-based information, which is a key factor that could enhance contextualisation by incorporating interactions among gene set members.

In order to address these issues and facilitate the analysis and interpretation of bulk RNA-seq data, we previously published a standardised workflow for bulk RNA-seq data analysis, promoting interactive and reproducible processes using the R packages developed in our group [Ludt et al., 2022]. This workflow provides a step-by-step documentation of a typical bulk RNA-seq data analysis, guiding users through the individual steps, showcasing best practices and promoting reproducibility. However, during our work, we recognised a significant gap in the workflow: the need for a tool specifically designed to enhance, simplify and standardise the interpretation of functional enrichment results.

While our previously developed package, `GeneTonic`, provides basic functionality for enrichment result exploration, it is certainly not tailored to this task. Consequently, many of our collaboration partners still resorted to the classical way of functional enrichment result interpretation: a manual inspection of the extensive list of results searching for patterns and interesting gene sets, which they further explored using databases such as the Gene Ontology (GO) or the Kyoto Encyclopedia of Genes and Genomes (KEGG). However, this process can be prone to bias, as familiar and expected gene sets may be easily recognised, while novel or unexpected findings might be overlooked, sometimes simply due to the sheer amount of available results.

In order to evaluate whether this need was only prevalent to our research group and collaboration partners or portrayed a larger issue within the scientific community, I conducted a literature review. The findings confirmed that inadequate documentation and reporting of functional enrichment methods are widespread across the scientific community, with over 75% of reviewed studies failing to properly detail their analysis, thus making it difficult to impossible for peers to verify and reproduce the results. Additionally, the review revealed that published studies frequently highlight gene sets simply because they appear at the top of the list of results due to their statistical significance. This could overall imply that the large lists of functional enrichment results are not fully studied, which could lead to important results and insights being lost.

As part of this thesis, I developed a tool which streamlines and simplifies the interpretation of functional enrichment results. These efforts are composed in GeDi, an R/Bioconductor package that aggregates gene sets into meaningful clusters based on various measures of (dis)similarity, thereby reducing redundancy and improving the clarity of the results. GeDi achieves this by implementing a suite of **Ge**ne set **Di**stance metrics and clustering algorithms. Additionally, GeDi integrates protein-protein interaction information into the analysis to provide a more comprehensive view of the biological processes at play.

GeDi supports interactive exploration and detailed drill-down analyses within its framework through an integrated Shiny application, while also allowing for seamless integration into existing workflows via its stand-alone functionality. By offering multiple entry points and accommodating a wide range of use cases, GeDi caters to a diverse audience, particularly relevant given the increasing volume of enrichment analyses being conducted. With interactive visualisations and result aggregations, GeDi not only reduces the time required for researchers to analyse and interpret results but also helps minimise bias that would be introduced by manual inspection, which is the current standard practice in many analyses. In doing so, GeDi facilitates a more efficient and objective interpretation of the data, as showcased in this thesis on publicly available bulk RNA-seq data.

With its functionality for interactive data exploration, flexible stand-alone features, and seamless integration into our standardised bulk RNA-seq workflow, GeDi aims to improve the reporting standards of functional enrichment analyses in published research. Additionally, GeDi promotes reproducibility through an automated report generation feature.

By making data interpretation more efficient, accessible and reproducible, GeDi has the potential to drive new research efforts and simplify the generation of novel hypotheses, ultimately advancing the field of omics analysis.

1 Introduction

Exploring the underlying mechanisms and deciphering the biological patterns associated with phenotypes, complex traits, or diseases remains a primary objective across a wide range of biological research domains [Gohlke et al., 2009; Cano-Gamez, Trynka, 2020]. Recent advancements in sequencing technologies, particularly high-throughput sequencing, have transformed our ability to study gene expression, making it possible to uncover intricate biological processes at an unprecedented level of detail [Trapnell et al., 2009; Wang et al., 2009; Reimand et al., 2019; Stark et al., 2019; Van Den Berge et al., 2019; Li, Wang, 2021]. These technological leaps have enabled researchers to generate extensive datasets that capture the complexity of biological systems.

At the same time, the sheer volume and complexity of omics data, such as genomics, transcriptomics and proteomics, present significant challenges for interpretation. Analysing and making sense of these datasets requires sophisticated tools capable of processing, integrating, and aggregating results while minimising redundancy [Girolami et al., 2006; Conesa et al., 2016]. This need is particularly pressing in the field of transcriptomics with a specific focus on functional enrichment analysis, where identifying biologically meaningful patterns within gene sets is key to understanding the molecular basis of various conditions [Subramanian et al., 2005; Reimand et al., 2019; Geistlinger et al., 2021].

In order to overcome these challenges, the development of efficient workflows and analytical tools has become essential. Streamlined workflows can significantly reduce the time spent manually sorting through large, redundant result sets and enhance the overall clarity of findings [Berger et al., 2013; Ritchie et al., 2015]. By simplifying the data interpretation process, these tools enable researchers to focus on extracting biologically relevant insights and formulating testable hypotheses.

Before exploring the challenges addressed in this thesis, it is important to establish a shared foundation of the underlying molecular mechanisms and terminology. A common understanding of these principles is necessary for contextualising both the complexity of omics data and the significance of the tools and methodologies designed to interpret them.

This groundwork will serve as a guide, leading to a deeper understanding of the advancements in the field and the innovative solutions developed to address its inherent challenges. Accordingly, the following sections will offer a broad introduction to the molecular basis of gene expression, setting the stage for a more detailed discussion on gene expression analysis and its complexities.

1.1 Molecular Foundations of Gene Expression

The genome represents the complete set of genetic information within an organism [Alberts, 2002]. The term genome was first introduced by Hans Winkler in 1920 as a blend of the words **gen**e and chromos**ome** [Winkler, 1920]. Since then, a variety of research fields emerged, further specifying and defining this term until it has reached its current definition.

In nearly all organisms, the genome is composed of deoxyribonucleic acid (DNA), which consists of two strands of nucleotides (Figure 1). Nucleotides are organic molecules consisting of a phosphate group, a deoxyribose sugar and one of four canonical nitrogenous bases: adenine (A), cytosine (C), guanine (G) or thymine (T). The two nucleotide strands run in antiparallel directions and are connected through base pairings, where adenine pairs with thymine (A-T) and cytosine pairs with guanine (C-G). This base pairing enables the DNA to form the well-known double-helix structure [Watson et al., 1953; Alberts, 2002; Alberts et al., 2014].

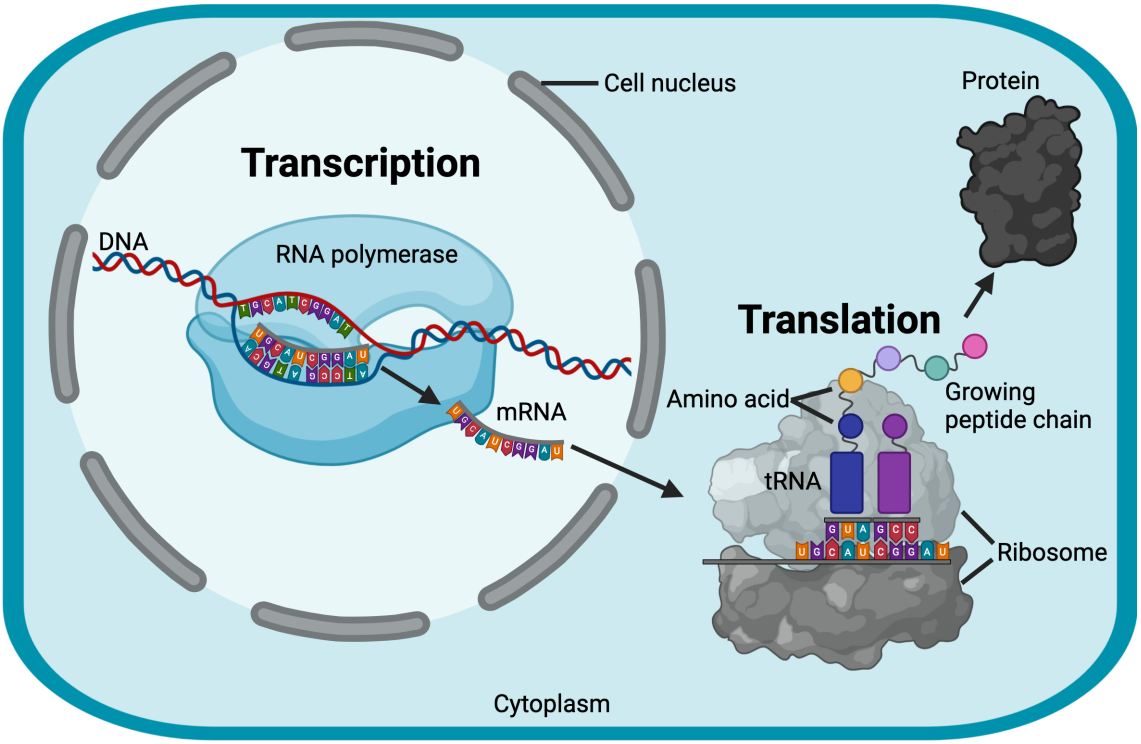


Figure 1 – The Central Dogma of Molecular Biology
The figure shows the information flow from DNA through RNA to the final protein. The figure shows a simplified depiction of a singular cell with its nucleus and cytoplasm. The processes of transcription and translation are shown. The figure was created using Biorender.com

The sequence of these base pairs encodes all the genetic information necessary for an organism's growth, development, functioning, and reproduction. Each DNA strand contains the same genetic information, ensuring the accurate replication of genetic material during cell division and other processes [Alberts, 2002; Alberts et al., 2014]. DNA not only serves as the blueprint for replication but also acts as a template for protein synthesis, where genes provide the instructions for producing proteins. However, genes represent only a small fraction of the genome — approximately 1-2% in the human genome [The ENCODE Project Consortium, 2011]. The remaining 98-99% consists of non-coding DNA, which can be further divided into regulatory sequences, introns, transposable elements, non-coding ribonucleic acid (RNA) and what was previously called "junk" DNA; a part of the genome that was assumed to be strictly non-coding, non-regulatory DNA which has since been revealed to be also of regulatory and structural importance [The ENCODE Project Consortium, 2011].

Genes play important roles in biological processes such as metabolism, cell division, and immune response [Alberts, 2002; The ENCODE Project Consortium, 2011; Alberts et al., 2014; Krebs et al., 2017]. These processes rely on the precise regulation and expression of genes. This process of gene expression generally involves two main steps: transcription and translation.

During transcription, DNA is transcribed into RNA by the enzyme RNA polymerase (Figure 1). Structurally, RNA is similar to DNA. However, RNA contains a ribose sugar instead of a deoxyribose sugar and the nitrogenous base thymine is substituted by uracil (U) [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017]. There are several types of RNA, including messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA). While tRNA and rRNA are non-coding RNAs essential for protein synthesis, mRNA is responsible for carrying the genetic information from DNA to the ribosome, where proteins are synthesised. During transcription, the RNA polymerase binds to the DNA, unwinds the double helix, and synthesises a complementary mRNA strand by adding RNA nucleotides that pair with the DNA template (Figure 1). Once the mRNA strand is complete, it detaches from the DNA, allowing the double helix to reform [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017].

After transcription, the mRNA may undergo post-transcriptional modifications that enhance its stability and facilitate the transport to the ribosome. These modifications protect the mRNA from degradation (e.g., 5' capping) or aid in the transport of the mRNA (e.g., 3' polyadenylation) [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017]. Additionally, in many organisms, splicing occurs where non-coding regions (introns) are removed from the mRNA, and coding regions (exons) are joined together. Through alternative splicing, different mRNA variants can be generated from the same gene, thereby increasing the diversity of proteins produced. Once fully processed, the mRNA is transported to the ribosome for translation [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017].

In the ribosome, a complex molecular machine composed of rRNA and proteins, the mRNA sequence is translated into an amino acid sequence. During this translation process, the ribosome reads the mRNA in triplets of nucleotides called codons, with each codon specifying a particular amino acid. The codons are recognised by tRNA molecules carrying the corresponding amino acid. The ribosome facilitates the binding of tRNA to mRNA and catalyses the formation of peptide bonds between adjacent amino acids, thereby extending the growing polypeptide chain. As the ribosome progresses along the mRNA, the sequence of codons is translated into a sequence of amino acids, which ultimately folds into a functional protein (Figure 1) [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017].

Like mRNA, proteins may undergo further modifications after translation. These post-translational modifications include the addition of functional groups such as phosphates (phosphorylation), carbohydrates (glycosylation), or lipids (lipidation), cleavage of specific peptide bonds, or the formation of disulfide bonds. These modifications are essential for ensuring that proteins adopt their functional three-dimensional (3D) structure. The

correct folding and modification of proteins are critical, as their 3D structure determines their ability to interact with other molecules and perform their specific biological roles [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017].

Once fully formed, proteins are the essential molecular machinery of the cell, performing a wide array of functions such as transport, signalling and structural support as well as catalysing chemical reactions in the form of enzymes [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017]. The sequence of a protein is determined by its corresponding gene, and its ultimate function is dictated by its 3D structure. This diversity of protein functions is essential to nearly every biological process that sustains life, making proteins central to the study of biology [Alberts, 2002; Alberts et al., 2014; Krebs et al., 2017].

Together, DNA, RNA, and proteins form a complex system that underlies nearly every cellular function. Even minor disruptions in this finely tuned machinery can cause significant shifts in an organism's overall functioning, potentially leading to severe outcomes such as disease or, in extreme cases, death [Gohlke et al., 2009; Dayalu, Albin, 2015; Zeng et al., 2021; Britto-Borges et al., 2022; He et al., 2023].

Therefore, understanding gene expression is essential across numerous research fields, as it provides key insights into the mechanisms that regulate both health and disease.

1.2 Gene Expression Analysis using RNA-Sequencing

In modern molecular biology, gene expression analysis is widely used to investigate the expression profiles of genes in cells, tissues, or entire organisms [Alberts et al., 2014; Raghavachari, Garcia-Reyero, 2018]. These analyses are used to examine the molecular mechanisms underlying various diseases or phenotypes, providing valuable insights into their origins. While proteins are central to biological processes and disease mechanisms, gene expression analysis is typically conducted at the RNA level. By examining the complete set of RNA transcripts produced by a genome, researchers can quantify gene expression in a given cell, tissue, or organism [Alberts et al., 2014; Raghavachari, Garcia-Reyero, 2018].

This collection of RNA transcripts, known as the transcriptome, represents the genes actively expressed at a particular time within a cell [Alberts et al., 2014; Van Den Berge et al., 2019]. As such, the transcriptome offers a snapshot of cellular functions at a given state, revealing how cells behave and respond to different environmental or experimental conditions. Through the analysis of the transcriptome, researchers can gain a comprehensive understanding of gene expression patterns, identify novel transcripts, and uncover regulatory mechanisms controlling gene expression [Alberts et al., 2014; Van Den Berge et al., 2019; Marini et al., 2021].

Transcriptome analysis began to gain prominence in the 1990s with the development of microarrays, one of the earliest high-throughput methods for studying gene expression [Schena et al., 1995; Graves, 1999; Blohm, Guiseppi-Elie, 2001; Raghavachari, Garcia-Reyero, 2018]. Although RNA had been studied since the 1970s using Northern blotting, this technique was extremely limited in terms of scale, as it could only analyse a few specific RNA molecules at a time [Alwine et al., 1977; Trayhurn, 1996]. In contrast,

microarrays allowed for the simultaneous measurement of thousands of genes, transforming transcriptomics by enabling large-scale gene expression analysis [Schena et al., 1995; Graves, 1999; Blohm, Guiseppi-Elie, 2001]. Thanks to this, microarrays quickly became the standard method for analysing gene expression, helping to uncover genes related to various biological processes and diseases. However, despite their advantages, microarrays required prior knowledge of gene sequences, limiting their ability to detect novel transcripts [Raghavachari, Garcia-Reyero, 2018].

By the early 2000s, RNA-sequencing (RNA-seq) revolutionised transcriptomics, providing a more precise, comprehensive, and high-throughput method for measuring gene expression. Unlike microarrays, RNA-seq did not require prior knowledge of gene sequences, making it ideal for discovering novel transcripts and identify alternative splicing. The technique also enabled the detection of non-coding RNAs, positioning RNA-seq as the current gold standard for transcriptome analysis, due to its versatility and depth of information [Wang et al., 2009; Raghavachari, Garcia-Reyero, 2018; Hong et al., 2020].

1.2.1 The Basics of RNA-Sequencing

RNA-sequencing is a powerful and widely-used technique for analysing the transcriptome, which is the complete set of RNA transcripts produced by the genome in a specific cell, tissue or organism at any given time. As such, RNA-seq is able to provide quantitative data across the entire transcriptome, offering a high-resolution view of the gene expression [Adiconis et al., 2013; Van Den Berge et al., 2019; Amezquita et al., 2020; Heumos et al., 2023]. It surpasses previous methods like microarrays by offering greater sensitivity, a broader dynamic range and the ability to detect novel transcripts without prior knowledge of the genome [Mortazavi et al., 2008; Wang et al., 2009].

There are two main approaches to RNA-seq: bulk RNA-sequencing and single-cell RNA-sequencing (scRNA-seq). In bulk RNA-seq, RNA from a large population of cells is pooled together before sequencing, producing an average gene expression profile across the entire cell population. In contrast, scRNA-seq isolates individual cells for sequencing, allowing for the analysis of gene expression at the single-cell level and capturing the heterogeneity among individual cells. With this increased granularity of the technique, scRNA-seq has advanced the study of complex tissues and rare cell types, expanding the versatility of RNA-based methods across a broader range of research fields [Stuart, Satija, 2019; Amezquita et al., 2020; Heumos et al., 2023]. Despite the increasing use of single-cell sequencing technologies, bulk RNA-seq remains a widely adopted method due to its cost-effectiveness, simplicity, and robustness [Stark et al., 2019; Li, Wang, 2021].

A typical bulk RNA-seq analysis involves four main steps: wet-lab preparation of the RNA, sequencing, data processing, and lastly, modelling and downstream analysis (Figure 2). The process typically begins with a research objective and the planning of the experiment. Once this foundation is set, the wet-lab preparation begins with RNA extraction from tissue or cell samples, ensuring that the RNA is carefully isolated to prevent degradation. Afterwards, the RNA is enriched, followed by fragmentation into suitable lengths used for sequencing. These fragmented RNA molecules are then reverse-transcribed into complementary DNA (cDNA) to which sequencing adapters are ligated. In a last step, the resulting, double-stranded cDNA is amplified using techniques such

as polymerase chain reaction (PCR) to generate a library for sequencing [Raghavachari, Garcia-Reyero, 2018]. A comprehensive overview of the wet-lab library preparation process can be found in the work by Kukurba, Montgomery [2015].

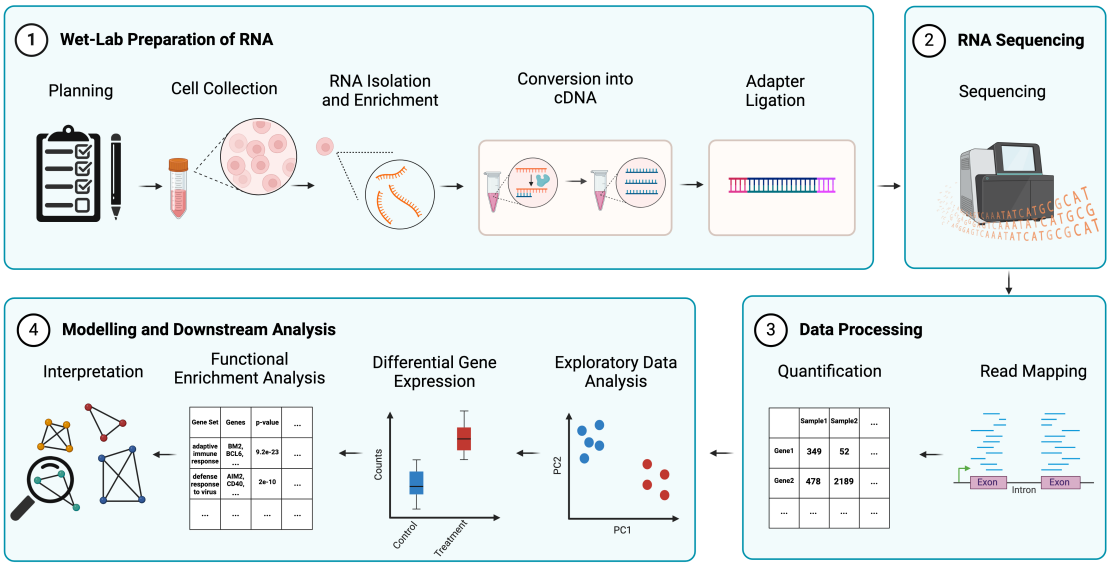


Figure 2 – A Typical Bulk RNA-Sequencing Analysis Workflow
This figure provides an overview of the bulk RNA-sequencing workflow, starting with the wet-lab preparation of RNA (**1**), which includes experimental planning, cell collection, RNA isolation and enrichment, conversion into cDNA and adapter ligation. This is followed by RNA sequencing (**2**), where the double-stranded cDNA is sequenced to generate raw reads. The reads are then aligned to a reference genome or transcriptome, followed by a quantification of the reads (**3**). Finally, during the modelling and downstream analysis (**4**), multiple analytical steps are applied, including exploratory data analysis, differential gene expression analysis and functional enrichment analysis, followed by interpretation of the data. The figure was created using Biorender.com

In the second step of the analysis, the cDNA library is sequenced, converting it into a FASTQ file containing raw reads and a quality score for each read. The quality score indicates the likelihood that a read has been correctly identified, helping to assess the overall accuracy of the sequencing (Figure 2) [Raghavachari, Garcia-Reyero, 2018].

In the third step of the workflow, the reads are aligned to a reference genome or transcriptome to determine their origin (Figure 2). This is done using tools like STAR [Dobin et al., 2013], HISAT2 [Kim et al., 2019], Salmon [Patro et al., 2017] or Kallisto [Bray et al., 2016]. Each tool has a specific approach to the alignment of the data, hence the choice should be based on the data at hand and the planned analysis [Dobin et al., 2013; Bray et al., 2016; Patro et al., 2017; Kim et al., 2019]. For example, tools such as STAR and HISAT2 implement accurate alignment algorithms, which determine the precise base-level position of each read on the genome or transcriptome [Dobin et al., 2013; Kim et al., 2019]. In contrast, Salmon and Kallisto use quasi-mapping and pseudo-alignment approaches, omitting the step of fully aligning each read. However, this does not reduce the accuracy of the resulting alignment, but is rather a step taken to prioritize the speed of the alignment [Bray et al., 2016; Patro et al., 2017].

After alignment, transcript expression levels (i.e., the amount of RNA produced by a specific gene) are quantified by counting the number of reads that align to each transcript. For accurate alignment methods (i.e., STAR and HISAT2), this involves counting

reads mapped to specific genes or transcripts, while in quasi-mapping approaches, transcript abundances are estimated probabilistically during the mapping step itself.

Once the read counts have been quantified, the modelling and downstream analysis of the data begins (Figure 2). This fourth step usually begins with read normalization to account for differences in sequencing depth or technical variations in the data [Yang, Kim, 2015; Conesa et al., 2016; Van Den Berge et al., 2019; Jiang et al., 2024]. This normalization is part of the exploratory data analysis (EDA), during which the quality of the samples is assessed and potential outlier samples are identified. Following the EDA, a differential gene expression (DE) analysis is performed. This step identifies genes which are significantly up- or downregulated between different experimental conditions [Yang, Kim, 2015; Conesa et al., 2016; Van Den Berge et al., 2019; Jiang et al., 2024]. Afterwards, the biological significance of these differentially expressed genes (DEGs) is assessed through functional enrichment analysis which identifies biological processes or pathways enriched in DEGs and reveals the functions altered between experimental conditions [Yang, Kim, 2015; Conesa et al., 2016; Van Den Berge et al., 2019; Jiang et al., 2024]. In a last and iterative step, the results of the analysis are subject to interpretation and contextualisation in order to draw conclusions from the experiment and generate hypotheses for new research questions.

1.2.2 RNA-Sequencing Data Analysis using R and Bioconductor

While numerous tools are available for the data processing step (Figure 2), the modelling and downstream analysis steps are typically conducted using the R programming language [R Core Team, 2024]. Initially developed in the early 1990s as a programming language for statistical computing and data analysis, R has grown into a versatile and robust framework widely used by researchers in the fields of statistics, data science and bioinformatics. R is open-source and available on all major operating systems, supporting a vast array of analyses.

A key strength of R is the extensive collection of packages which serve as extensions that enhance the base functionality of the language. Many of these packages are available via an official repository archive called CRAN (The Comprehensive R Archive Network, `https://cran.r-project.org`) [R Core Team, 2024]. At the time of writing, over 20,000 packages are available on the platform, providing implementations for statistical and machine learning methods, various visualisation options and specialised data analysis packages [R Core Team, 2024].

Besides CRAN, Bioconductor has emerged as one of the largest repositories of freely available R packages. The Bioconductor project (`https://bioconductor.org`) is an open-source project focusing on packages tailored to the analysis of biological data [Gentleman et al., 2004; Huber et al., 2015; Amezquita et al., 2020]. In addition to software packages, Bioconductor also includes packages offering annotation and experimental data, complete step-by-step data analysis workflows as well as extensive documentation and tutorials. Lastly, GitHub (`https://github.com`) is a widely used platform for hosting R packages, offering convenient storage alongside features for version control and enhanced collaboration. However, it is important to note that this is by no means an exhaustive list of R package sources.

These resources provide an extensive number of R packages for the individual steps of data processing and modelling, making R the preferred programming language for bulk RNA-sequencing analysis (Figure 2). Thanks to the efforts of the R and Bioconductor communities, researchers have access to a vast collection of packages that enable them to construct workflows tailored to their specific analyses. In the following, I will introduce some of the most commonly used package options to demonstrate how an RNA-seq analysis workflow can be effectively assembled using R and Bioconductor.

For the quantification of the sequencing reads into count data, the `Rsubread` package is commonly used [Liao et al., 2019], though other approaches and implementations exist [Li, Dewey, 2011; Trapnell et al., 2012; Anders et al., 2015; Pertea et al., 2016]. The result of the quantification is a matrix of counts, representing the expression of each feature (usually a gene) in the data [Liao et al., 2019]. The simple approach implemented in the `Rsubread` package is to record the number of reads which are overlapping the exons of a gene. Despite its simplicity, the approach has proven robust and produces reliable results for downstream analysis steps [Liao et al., 2019]. In contrast, as stated above, quasi-mapping methods like those implemented in Salmon and Kallisto provide an alternative approach to quantification by estimating the number of reads mapping to a specific exon directly without a full alignment step. These tools generate transcript-level quantifications quickly, which can then be imported and summarised to the gene level in R using packages like `tximport` [Soneson et al., 2015; Patro et al., 2017]. This offers a more flexible and computationally efficient pipeline for handling large RNA-seq datasets.

The count data should afterwards be normalised to ensure that the expression measurements are comparable across samples and conditions. This should include adjustments for varying sequence depth and library composition across samples. For this step, a number of tools and approaches are available, which have been discussed and evaluated in the work of Dillies et al. [2013] and Evans et al. [2018]. In the R user community, the normalization approaches implemented in the `DESeq2` [Love et al., 2014], `edgeR` [Robinson et al., 2010; Chen et al., 2024] or `limma` [Ritchie et al., 2015] package are widely used.

Before proceeding with the differential gene expression analysis, it is important to perform exploratory data analysis to assess the data quality and identify potential issues such as outliers, batch effects, or technical variability (Figure 2). This step ensures that the subsequent analyses are based on high-quality, well-prepared data. Techniques like Principal Component Analysis (PCA) are commonly employed to visualise how samples are distributed and whether they cluster according to the biological conditions of the data [Mead, 1992; Jolliffe, 2002; Jollife, Cadima, 2016]. PCA can help reveal whether technical factors, such as batch effects, are influencing the data, while visualisations such as heatmaps and sample correlation plots further clarify relationships between samples, helping to detect potential outliers [Anders et al., 2015; Conesa et al., 2016; Marini, Binder, 2019]. In R, a comprehensive tool for exploratory data analysis is `pcaExplorer`, an interactive package to perform PCA and further quality control analyses on the data [Marini, Binder, 2019].

In the next step of the downstream analysis, differentially expressed genes are identified in order to understand the differences in gene expression between samples (Figure 2). As this is one of the most common steps in an RNA-seq data analysis, numerous tools

have been implemented and refined over the years. These tools are designed to handle different experimental designs, sample sizes, and dataset-specific challenges, allowing researchers to select the method best suited to their analysis. Soneson and Delorenzi [Soneson, Delorenzi, 2013] have published an extensive comparison of DE analysis tools, highlighting that for most cases — especially when the sample size is small — the implementations in the packages `DESeq2` [Love et al., 2014] and `edgeR` [Robinson et al., 2010; Chen et al., 2024] show a favorable balance between true positive and false positive rates in their results [Soneson, Delorenzi, 2013; Schurch et al., 2016]. Additionally, the approach implemented in the `limma` package [Ritchie et al., 2015] has shown good performance in identifying DEGs despite being originally designed for microarray analysis, provided that the counts have been preprocessed using the voom transformation of the package [Ritchie et al., 2015].

As the foundation of many diseases and pathways is usually a complex interplay of a variety of genes, the set of differentially expressed genes is typically not the final result of a bulk RNA-seq analysis (Figure 2). Once DE genes have been determined, their biological function and role in the sample at hand is analysed using functional enrichment analysis. This analysis can identify the pathways and biological functions in which the DEGs are involved, reducing the sometimes extensive amount of genes and providing biological context. This can greatly enhance and facilitate interpretation of the results and provide a foundation for future studies and experiments.

1.3 Functional Enrichment Analysis

Functional enrichment analysis aims to identify biological functions or pathways that are over-represented or significantly enriched within a set of genes or proteins which share common features, such as differential expression [Geistlinger et al., 2021; Garcia-Moreno et al., 2022; Wijesooriya et al., 2022]. There are several approaches to functional enrichment analysis, which are generally grouped into three main categories [Khatri et al., 2012]: Over-Representation Analysis (ORA), Functional Class Scoring (FCS) and Pathway Topology (PT)-based methods. In ORA methods, the functional categories or pathways are assessed for an over-representation of the differentially expressed genes compared to a set of background genes [Khatri et al., 2012; Geistlinger et al., 2021; Wijesooriya et al., 2022]. FCS methods, in contrast, analyse the distribution of DEGs within a (p-value or fold change) ranked list of the entire transcriptome, while PT-based methods incorporate additional pathway structure information to evaluate the impact of expression changes in genes and proteins [Ma et al., 2019].

In the broader scientific landscape, Over-Representation Analysis methods have been widely adopted due to them being generally considered as fast and reliable [Geistlinger et al., 2021; Wijesooriya et al., 2022]. Even though PT-based methods may offer a more nuanced analysis, the prevalence of ORA approaches highlights their practical utility and effectiveness in various research endeavours. Thus, given the extensive range of workflows based on these methods, this thesis will focus on the general characteristics and results of ORA methods.

A typical ORA approach involves three steps. First, a reference set of genes or proteins is selected which represents the background or universe of all genes or proteins which are

potentially relevant to the biological question at hand. This could, for instance, include all genes or proteins expressed in the dataset under study. Second, the set of differentially expressed genes or proteins is identified based on a significance measure (e.g., adjusted p-values from differential expression analysis). Finally, statistical tests, such as hypergeometric or chi-square tests, compare the functional annotations of the selected genes or proteins with the background set. This comparison identifies functional categories that are significantly over-represented compared to what would be expected by chance [Khatri et al., 2012; Wijesooriya et al., 2022].

Among the variety of R packages available for functional enrichment analysis, `topGO` [Alexa et al., 2006] and `clusterProfiler` [Wu et al., 2021] are two of the most commonly used tools. `topGO` offers an ORA implementation focused on the Gene Ontology (GO), a bioinformatics database that stores information on gene functions and interactions in a hierarchical structure, from broad categories like "cellular process" to specific terms like "negative regulation of neuroinflammatory response" [Ashburner et al., 2000; Aleksander et al., 2023]. `clusterProfiler`, on the other hand, supports analyses of a more extensive range of databases [Wu et al., 2021]. These databases include, but are not limited to, the Kyoto Encyclopedia of Genes and Genomes (KEGG) [Kanehisa et al., 2017, 2019], the Reactome database [Fabregat et al., 2018] and the Molecular Signatures Database (MSigDB) [Liberzon et al., 2011, 2015].

Thanks to advancements in technology and analysis methods, transcriptome as well as functional enrichment analysis have become more and more accessible to bioinformaticians and computational biologists. However, with the increasing amount of bulk RNA-seq datasets generated, molecular biologists and other researchers often face the challenge of analysing these datasets computationally, despite having limited or no prior experience in programming and data analysis, with both skill sets being a prerequisite to these tasks. While external experts can solve this bottleneck, not all research groups have the necessary resources to hire fully trained bioinformaticians, leaving existing team members with the need to acquire these essential skills themselves on top of their existing expertise.

While excellent standardised protocols and various interactive, user-friendly tools exist for bulk RNA-seq data analysis, as demonstrated in our manuscript "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022], the interpretation of the results remains a significant challenge in individual analyses. During the preparation of our manuscript, we identified a significant gap in the availability of interactive tools specifically designed for interpreting functional enrichment results. Despite enrichment analysis being a common step in bulk RNA-seq analysis, most of the available tools present results as long, tabular lists that can often include redundant gene sets, making it difficult to efficiently analyse and interpret the data (Figure 3) [Huang et al., 2009; Merico et al., 2010; Supek et al., 2011; Supek, Škunca, 2017; Yoon et al., 2019].

GO.ID	Term	genes	p.value_elim	p.value_classic	Annotated	Significant	Expected
GO:0051301	cell division	Aatf,Actr2,Actr3,Ahctf1,Akna,Alkbh4,Anapc10,Anapc1	1.1e-09	8.9e-16	603	288	194.92
GO:0043065	positive regulation of apoptotic process	Acin1,Acvr1c,Adam10,Adam17,Adam8,Adora2a,Adrb2	1.7e-09	9.2e-12	610	276	197.19
GO:0043066	negative regulation of apoptotic process	Aatf,Acvr1,Ada,Adam17,Adam8,Adar,Adnp,Adora2a,Ag	1.1e-06	3.1e-09	926	382	299.33
GO:0002726	positive regulation of T cell cytokine production	Arid5a,B2m,Cd55,Cd55b,Cd81,Fzd5,Gata3,H2-D1,H2-	5.3e-06	5.4e-06	34	24	10.99
GO:0006364	rRNA processing	Bms1,Bud23,Bysl,C1d,Cdkn2a,Ddx10,Ddx27,Ddx54,D	1.1e-05	2.5e-08	214	108	69.18
GO:0006886	intracellular protein transport	Adar,Agk,Ahcyl1,Akt1,Anp32b,Ap1b1,Ap1m1,Ap1m2,A	1.4e-05	1.1e-10	648	286	209.47
GO:0048661	positive regulation of smooth muscle cell proliferation	Ager,Aif1,Akt1,Bmpr1a,Calcrl,Camk2d,Ccl5,Ccn4,Cyb	2E-05	2.1e-05	113	58	36.53
GO:0000079	regulation of cyclin-dependent protein serine/threonine kinase activity	Adam17,Akt1,Apc,Casp3,Ccna2,Ccnb1,Ccnb2,Ccnd3,	2.5e-05	2.6e-05	62	36	20.04
GO:0007265	Ras protein signal transduction	Arl6,Cnksr1,Csf1,Dab2ip,Dhcr24,Dok1,Dok2,Dok4,Erb	2.5e-05	2.6e-05	104	54	33.62
GO:1904951	positive regulation of establishment of protein localization	Aacs,Acsl3,Acsl4,Actr3,Adam9,Adora2a,Akt2,Ang,Ank	3.3e-05	5E-10	332	161	107.32
GO:0006915	apoptotic process	1600014C10Rik,Aatf,Acin1,Acsl5,Actn4,Acvr1,Acvr1b,	4.6e-05	5.4e-24	1909	816	617.1
GO:0033365	protein localization to organelle	4933427D14Rik,Abhd17a,Abhd17b,Abhd17c,Adam10,	5.1e-05	5.3e-13	945	408	305.48
GO:0050821	protein stabilization	Apbb1,Apbb2,Apoa2,Atf7ip,Atp1b1,Atp1b3,Bag1,Bag2	5.5e-05	5.8e-05	197	90	63.68
GO:0010628	positive regulation of gene expression	6030468B19Rik,Actg1,Acvr1b,Adam17,Adam19,Adam	5.8e-05	1.7e-11	1202	495	388.55
GO:0032729	positive regulation of type II interferon production	Arid5a,Ccr2,Cd160,Cd2,Cd244a,Cd3e,Cd40lg,Crtam,F	5.9e-05	6.2e-05	92	48	29.74
GO:0006954	inflammatory response	Abcc1,Abcd1,Abhd12,Abr,Acp5,Acvr1,Ada,Adam17,Ac	7E-05	1.6e-09	783	331	253.11
GO:0046112	nucleobase biosynthetic process	Ada,Aprt,Cad,Cmpk1,Ctps1,Ctps2,Dhodh,Gart,Hprt1,F	7.9e-05	8E-05	16	13	5.17
GO:0090316	positive regulation of intracellular protein transport	Anp32b,B3gat3,Bag3,Bcap31,Brca1,Cd81,Cdc42,Cdk	8.1e-05	8.4e-05	105	53	33.94
GO:0072659	protein localization to plasma membrane	Abi3,Acsl3,Actn2,Actr3,Adcy6,Afdn,Akt1,Akt2,Ank3,An	9.5e-05	2.4e-08	311	147	100.53

Figure 3 – A Typical Functional Enrichment Result
The figure shows a section of a typical functional enrichment result table. The figure shows the first 20 of the originally 500 gene sets in the result.

Typically, researchers manually sift through these lists in search of interesting gene sets and patterns. This approach is not only cumbersome and time-consuming but also forces researches to keep track of a vast amount of information by analysing numerous gene sets simultaneously. Clearly, this can lead to biased results and a lot of information being buried and lost among the plethora of possible interpretations and hypotheses. In their 2022 article, authors Wijesooriya et al. found a lacking standard in the documentation of functional enrichment results in published scientific work, with over 95% of articles inadequately documenting the conducted functional enrichment analysis, making it difficult to nearly impossible to reproduce the presented results [Wijesooriya et al., 2022]. Additionally, 43% of the investigated articles did not perform proper correction for multiple hypotheses testing, increasing the probability of false positive hits among the results, thus further reducing their reproducibility.

The article by Wijesooriya et al. [2022] as well as our own experience during the development of our manuscript [Ludt et al., 2022] highlighted the pressing need for an interactive tool which streamlines and simplifies the interpretation of enrichment results. Given this objective, I developed the R/Bioconductor package `GeDi`[1]. `GeDi` aims to achieve a more compelling aggregation of the detailed and sometimes overwhelming functional enrichment analysis results. This is accomplished by first calculating **Ge**ne set **Di**stance scores for the individual gene sets in the enrichment results, before these results are subsequently further aggregated through clustering on the calculated distances.

`GeDi` offers a variety of distance scores and clustering methods, allowing researchers to tailor the analysis to their specific needs. Moreover, `GeDi` incorporates biological network data, such as protein-protein interactions (PPI), to enrich the analysis with added biological context, thereby enhancing the interpretability of functional relationships among gene sets. All package functionality is available as stand-alone functions or within the `Shiny` web application, facilitating interactive exploration and aggregation of functional enrichment analysis results. The required input is the result of a preceding functional

[1] GeDi is a portmanteau combining the terms **Ge**ne set and **Di**stance.

enrichment analysis, which can be provided in plain text, spreadsheet formats, or standardised formats commonly used in the R/Bioconductor community [Gentleman et al., 2004; Huber et al., 2015; R Core Team, 2024].

`GeDi` is embedded in the R/Bioconductor package ecosystem [Huber et al., 2015; R Core Team, 2024], allowing for the use of various different functional enrichment approaches implemented in different R packages such as `topGO` or `clusterProfiler` [Alexa et al., 2006; Wu et al., 2021]. This integration makes `GeDi` an ideal tool to be incorporated into various analysis workflows, including those for bulk and single-cell RNA-seq [Amezquita et al., 2020]. By summarising and visualising the often overwhelming amount of information contained in enrichment results, `GeDi` allows for a more streamlined interpretation and the generation of new hypotheses. The package is freely available under the MIT license on Bioconductor (`https://bioconductor.org/packages/GeDi`), with the development version accessible on GitHub (`https://github.com/AnnekathrinSilvia/GeDi`).

1.4 Aim of this Thesis

With the continuing advancement of high-throughput sequencing, the amount of generated bulk RNA-sequencing data has grown substantially, as the method becomes a standard component of numerous studies across various research fields. As the volume of generated data has been increasing, so has its complexity, necessitating robust methods to ensure a proper analysis according to current best practices. In this context, it is especially important to ensure the reproducibility of the conducted data analysis, in order to foster transparency and accuracy as well as to drive progress in the scientific community.

In this thesis, I aim to develop a workflow to enhance the analysis and interpretation of transcriptome data by leveraging the suite of R packages developed within our group. The goal is to develop a unified and standardised workflow for the analysis of bulk RNA-seq data. This workflow was previously published in a manuscript, which provides step-by-step, well-documented guidelines on how to use and integrate these R packages into a standardised workflow [Ludt et al., 2022]. Throughout this thesis, I will summarise key aspects of that manuscript, emphasising the importance of reproducible data analysis. Additionally, I will demonstrate the application of this workflow on a published RNA-seq dataset, in order to highlight its ability to improve data analysis and interpretation.

To further emphasise the need for reproducible analysis and detailed documentation on the results reported in existing works, I will also conduct a literature review of published articles involving functional enrichment results. This review is designed to assess the reproducibility of the reported methods and explore how functional enrichment results are generally presented in the literature. As functional enrichment analysis is an essential step of a typical RNA-seq analysis, which is not yet fully covered by our existing suite of R packages, the review also aims to identify how urgent the need for proper tools for the interpretation of these results is.

Hence, in order to extend our workflow by an additional, and certainly important, component, a key objective of this thesis is the development of a tool that streamlines and facilitates the interpretation of functional enrichment results. While numerous tools and implementations are available to calculate these results, these methods often produce extensive lists of gene sets, making it difficult to synthesise findings due to redundancy. Furthermore, most current approaches do not consider network-based information, which can provide additional biological context by incorporating gene set interactions.

For this purpose, I will present a new R package called `GeDi`, designed to address these challenges of functional enrichment analysis interpretation. `GeDi` aggregates gene sets into meaningful clusters based on their similarity, reducing redundancy and clarifying results. This is achieved through the implementation of various **Ge**ne set **Di**stance metrics and clustering algorithms. Additionally, the package includes a `Shiny` application, which allows for the interactive exploration of the data. Using publicly available murine bulk RNA-seq data, I will demonstrate `GeDi`'s functionality showcasing its potential to improve the interpretation of functional enrichment analysis results by providing a more efficient and objective interpretation of the data.

1.5 Structure of this Thesis

Chapter 1 provides the essential biological background necessary to understand the work presented in this thesis. The chapter opens with an explanation and discussion of the genome, transcriptome and proteome, focusing on their relationships and key differences (Section 1.1). Subsequently, Section 1.2 discusses gene expression analysis, specifically bulk RNA-sequencing, outlining the analysis from the process of data generation to the data analysis and interpretation. The core principles of the technique are discussed in Section 1.2.1, while Section 1.2.2 highlights the tools and packages commonly employed for data analysis in this field. Finally, Section 1.3 introduces functional enrichment analysis and discusses the current lack of an exploration package for these types of analyses.

Chapter 2 covers the motivation, implementation choices and guiding principles behind the development of `GeDi`. In Section 2.1, the objective is to examine the current standards for functional enrichment documentation in published research and assess the need for a comprehensive exploration and interpretation tool. In order to achieve this, I conduct a thorough literature review. Following this, Section 2.2 discusses the reasoning behind the choice of gene set distance metrics as a measure to quantify the (dis)similarity of gene sets. This section introduces the various distance metrics I have chosen to implement in `GeDi`, highlighting their area of application and providing mathematical definitions. Next, Section 2.3 details the clustering algorithms chosen for the analysis. In Section 2.4, I discuss the functionality of `GeDi`, with a particular focus on the data preparation and processing (Section 2.4.1) as well as the implementation of the distance metrics (Section 2.4.2) and their visualisations (Section 2.4.3) as well as the clustering algorithms (Section 2.4.4) and their corresponding visualisations (Section 2.4.5). The chapter concludes with Section 2.5, which introduces the `Shiny` application included in my package, emphasising its user interface and interactive features.

Chapter 3 presents a comprehensive use case of GeDi showcasing the functionality inherent in the package. In Section 3.1, the chapter opens with the introduction of a standardised bulk RNA-sequencing data analysis workflow, which I have previously published in Ludt et al. [2022]. This is followed by a discussion of the literature review findings, highlighting the current documentation standard of functional enrichment results. Afterwards, I demonstrate in Section 3.3 how functional enrichment results can be processed with GeDi to enhance the interpretability of a publicly available dataset. In Section 3.3.1, I introduce the dataset incorporated in the use case for the GeDi package, followed by Section 3.3.2 and Section 3.3.3, which describe a standardised workflow for preprocessing this dataset for its use with GeDi. Lastly, Section 3.3.4 includes the functional enrichment results interpretation of the data to showcase the functionality of GeDi and highlight how the package can streamline and facilitate the interpretation.

Lastly, in Chapter 4, I reflect on the relevance of the findings presented throughout this thesis. The chapter begins by emphasising the importance of standardised analysis workflows (Section 4.1). Furthermore, Section 4.2 touches upon the results of the literature review, highlighting how GeDi could influence the current standard of method and result reporting in published literature. Afterwards, Section 4.3 recapitulates the design decisions I made throughout the development of GeDi and compares my package to existing tools, while Section 4.4 highlights the most important findings of the data analysis included in this thesis. In the latter parts of this chapter, I touch upon the limitations of the work presented in this thesis and provide an outlook on potential future developments and extensions for GeDi.

2 Methods and Implementation

The primary goal of this thesis is to develop a tool that facilitates the exploration and interpretation of functional enrichment analysis results in order to improve hypothesis generation from the data. Functional enrichment analysis is an essential step in a variety of data analysis workflows, including bulk and single-cell RNA-sequencing, as well as other omics analyses [Huang et al., 2009; Wijesooriya et al., 2022]. Given the growing number of such datasets, there is a pressing need for an effective way to summarise and visualise the large number of gene sets included in these results [Huang et al., 2009; Yoon et al., 2019; Wijesooriya et al., 2022].

In order to bridge this gap, I developed the R/Bioconductor package `GeDi`. `GeDi` is an interactive tool, designed to streamline and facilitate the exploration and interpretation of functional enrichment results through compelling aggregations and visualisations of the data. This is accomplished by first calculating **Ge**ne set **Di**stance (**GeDi**) scores for the individual gene sets in the enrichment results, followed by clustering these results based on the calculated distances. Instead of requiring users to analyse numerous gene sets simultaneously and keeping track of a vast amount of information, `GeDi` reduces the complexity by grouping similar gene sets into a smaller number of clusters. This aggregation allows for easier pattern recognition within the data, thus facilitating interpretation and supporting hypothesis generation.

Many of the package's functionality is available as stand-alone functions, allowing seamless integration into existing workflows. Additionally, `GeDi` offers a `Shiny` web application that allows for interactive exploration and interpretation of enrichment results. A demo version of the application is also available at `http://shiny.imbei.uni-mainz.de:3838/GeDi`, giving users the opportunity to explore its functionality firsthand and familiarise themselves with its features in a hands-on environment.

The following sections of this chapter will begin with a description of the literature review (Section 2.1) which I conducted to assess the current reporting standards in functional enrichment analysis. This research formed the foundation for identifying gaps in existing tools and motivating the development of `GeDi`. Subsequently, the chapter will explain the rationale behind the use of gene set distances as a core approach, followed by an introduction to the specific distance metrics implemented (Section 2.2) and the clustering algorithms implemented in `GeDi` (Section 2.3). In Section 2.4, I will describe the functionality available within `GeDi`, including the implementation of data preparation (Section 2.4.1), the distance metrics (Section 2.4.2) and their visualisations (Section 2.4.3), and the implementation of the clustering methods (Section 2.4.4) and the corresponding visualisations (Section 2.4.5). Finally, in Section 2.5, I will provide an overview of the `Shiny` application included in my package, which offers users an interactive platform for exploring and interpreting functional enrichment analysis results alongside tools which ensure the reproducibility of the conducted exploration.

2.1 Literature Review of Functional Enrichment Analysis in Scientific Publications

In their work, Wijesooriya et al. [2022] reviewed over 180 published articles presenting functional enrichment analysis results, aiming to assess how thoroughly these analyses were documented. Their findings were concerning, showing that 95% of the articles failed to properly document their methods, while over 40% did not include details on p-value corrections, suggesting the possibility that many of these results could contain false positives. The authors called for the development of open-access standards to improve both the research quality and reproducibility of functional enrichment analysis [Wijesooriya et al., 2022].

Although not the primary focus of their original work, the findings of Wijesooriya et al. have important implications for the quality of result reporting and interpretation in functional enrichment analysis. In order to investigate these aspects in published research, I conducted an independent literature search and review. The primary goal was to evaluate the current state of functional enrichment reporting, with a particular focus on transparency and reproducibility in recently published literature. In contrast to the work of Wijesooriya et al., which focused specifically on the methodologies and how they were executed, this review focused more on the interpretation and presentation of enrichment results in order to determine whether a tool like `GeDi`, which offers interactive exploration and clustering of results, could significantly enhance the interpretation and presentation of these analyses. By providing these features, `GeDi` may address critical gaps in current reporting standards. In order to support this evaluation, a total of 97 articles were selected and reviewed using a random sampling method (Figure 4).

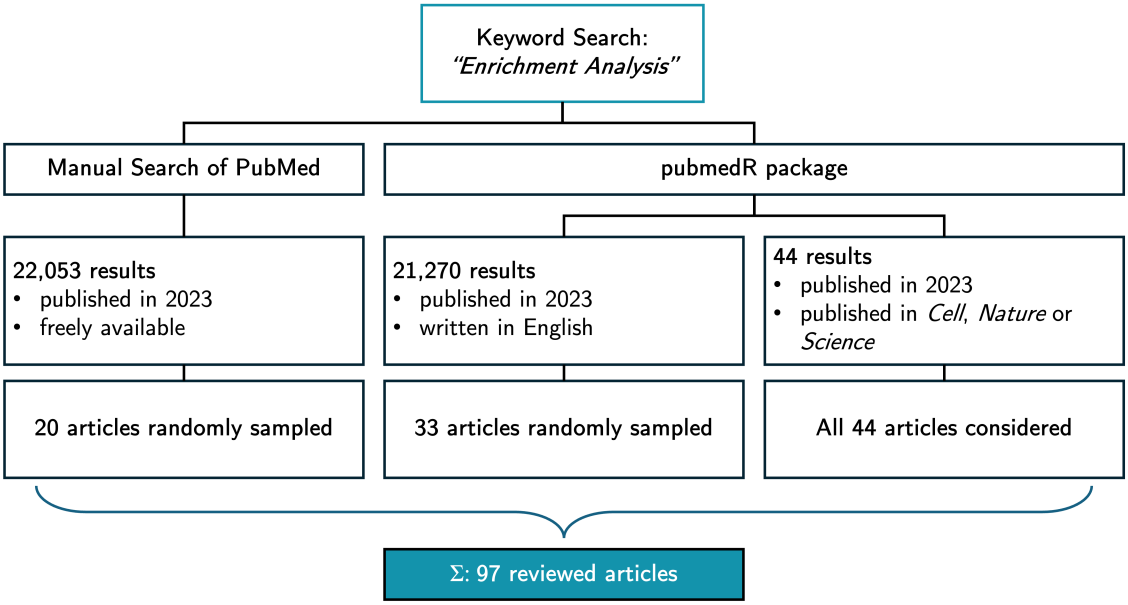


Figure 4 – Flowchart of the Literature Review
The figure shows the flowchart of the literature selection process, in which the keyword phrase "Enrichment Analysis" was used for three different rounds of literature search. Each round of literature review had slightly distinct criteria. In the end, 97 articles were used for the review.

The first round of literature review was performed using PubMed (https://pubmed.ncbi.nlm.nih.gov/, latest access: September 11th, 2024) using the keyword phrase "Enrichment Analysis". Additionally, the search was restricted to freely available articles published in 2023. This search returned 22,053 results, of which the first 10,000 were downloaded. With the use of R, the list of search results was processed and 20 articles were randomly selected for review.

Several parameters were evaluated during the review process. First, it was noted whether the article reported a functional enrichment analysis as described in Section 1.3, and if so, whether the methods were well documented. "Well documented" in this case was defined as providing the necessary amount of details in order for the reader to be able to accurately reproduce the results presented in the article. Additionally, I documented which gene set libraries and statistical methods or software were used, and the formats chosen to present the results, for example, tables, heatmaps or other visualisations. Other parameters evaluated included highlighting of specific subsets of results, the reasoning behind subset selections, whether results were aggregated (e.g., grouped or clustered), and any mentions of follow-up experiments, code availability, or limitations.

In the second round of literature review, the R package pubmedR [Aria, 2020] was used to ensure a reproducible selection of articles. As search scope, the same keyword phrase and restrictions of the first search were used, but it was additionally limited to articles published in English. Of the 21,270 articles returned, the first 9,999 were downloaded (limited by the download capacity of the pubmedR package [Aria, 2020]). Afterwards, articles were randomly selected and screened until a total of 20 articles presenting functional enrichment results were accumulated. As not all of the found articles presented enrichment results following the lines of Section 1.3, 33 articles were reviewed overall (Figure 4).

Finally, I performed a third literature search using the pubmedR package, this time focusing on articles published in the prestigious journals *Cell*, *Nature*, and *Science*. These journals are commonly known for their rigorous peer reviews and high standards of methodological transparency, making them ideal for reflecting the overall state of functional enrichment reporting in high-impact scientific publications. This search returned 44 articles, all of which were screened for the previous parameters (Figure 4).

In order to ensure reproducibility, I created a GitHub repository (https://github.com/imbeimainz/HIPSTER), which includes the initial PubMed search results, the code for selecting a reproducible subset of articles to investigate and the structured literature search using the pubmedR package [Aria, 2020]. In order to account for the factor of randomness, the seed of the random number generator which was used for the sampling, is also provided transparently in the repository, ensuring full reproducibility.

2.2 Definition of the Gene Set Distance Scores

As functional enrichment analyses constitute a central step in the analysis of various omics datasets, numerous tools and implementations are available in the R and Bioconductor community [Alexa et al., 2006; Huber et al., 2015; Wu et al., 2021; R Core Team, 2024]. However, despite their utility, existing methods often yield outputs com-

prising extensive lists of gene sets, impeding hypothesis generation and synthesis due to inherent redundancy. In a typical workflow, users manually inspect individual gene sets, often requiring additional research in corresponding databases. This process is not only time-consuming and cumbersome, but also requires users to manage large amounts of information simultaneously. Consequently, valuable insights may get lost, and interpretation may become biased, as users tend to recognise and remember gene sets they are already familiar with more easily.

One approach to improve data interpretation and reduce redundancy of the data is an appropriate aggregation of the gene sets into groups of similar characteristics. This method can significantly simplify data interpretation by shifting the focus from analysing individual gene sets and their interrelationships to examining clusters of related gene sets. This reduction in complexity allows for a more streamlined and coherent analysis of the data.

The aggregation of gene sets into groups is achieved by using gene set distance scores. These scores quantify the (dis)similarity between individual sets allowing them to be aggregated into groups. In this context, a variety of different distance scores can be applied with varying focus and definitions of similarity between gene sets. For `GeDi`, six different distance scores were selected and incorporated into the final tool:

- Meet-Min Distance Score
- Jaccard Distance Score
- Kappa Distance Score
- Protein-Protein Interaction-Weighted Meet-Min Distance Score
- Sørensen-Dice Distance Score
- GO Semantic Distance Score

2.2.1 Meet-Min Distance Score

The Meet-Min (MM) distance score is used in biological research to evaluate the functional similarity between two sets such as gene sets or pathways [Yoon et al., 2019]. This metric provides a quantitative measure of similarity between two sets by considering both the overlap and the relative sizes of the sets. It is based on the overlap coefficient [Vijaymeena, Kavitha, 2016], which compares the overlap between two sets while taking the size of the sets into account. The Meet-Min distance score is then defined as

$$\mathrm{MM}(A, B) = 1 - \frac{|A \cap B|}{min(|A|, |B|)}$$

where A and B are two distinct gene sets, respectively, and $|\cdot|$ represents their cardinality (i.e., the number of elements in the gene set).[2]

[2] This notation will be used throughout Section 2.2

The MM score is symmetric (i.e., $\mathrm{MM}(A, B) = \mathrm{MM}(B, A)$), and the range of the resulting distances is $[0, 1]$. A distance of 0 indicates that the sets are completely identical, while a distance of 1 reflects completely disjoint sets.

The Meet-Min distance score can be applied in scenarios where understanding the uniqueness and overlap between two sets is essential. As the MM score effectively normalises the intersection by the size of the smaller set, it can provide a more balanced view of similarity especially when comparing sets of different sizes. With this characteristic, it is frequently used when studying genomic intervals and their overlap; a field where accounting for the size of the two compared sets is of great importance [Ma et al., 2023].

2.2.2 Jaccard Distance Score

The Jaccard distance score is a measure of dissimilarity between two sets based on the commonly applied Jaccard index [Jaccard, 1912; Levandowsky, Winter, 1971]. The Jaccard index measures the similarity between two sets by dividing the size of their intersection by the size of their union. The distance measure of dissimilarity is then defined as

$$\mathrm{Jaccard}(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard distance score is symmetric (i.e., $\mathrm{Jaccard}(A, B) = \mathrm{Jaccard}(B, A)$), and the range of the resulting distances is $[0, 1]$. A distance of 0 indicates that the sets are completely identical, while a distance of 1 reflects completely disjoint sets.

The Jaccard similarity and distance score are commonly used measures in various fields which is likely based on their straightforward interpretation as they directly reflect the ratio between shared and unique elements in both sets. As such, the Jaccard score can, for example, be used to analyse the presence and absence of species across different regions [Chung et al., 2019].

2.2.3 Kappa Distance Score

The Cohen's Kappa distance score, hereafter referred to as Kappa distance score, is derived from the equally named Cohen's Kappa coefficient. Initially, the coefficient was introduced to measure inter-rater reliability for categorical items [Cohen, 1960]. Since its first introduction, the Kappa distance score has been widely used as a measure of (dis)similarity between gene sets, e.g., in the work by Yoon et al. [2019] or the Database for Annotation, Visualisation, and Integrated Discovery (DAVID) [Huang et al., 2007, 2009].

As a distance measure, the Kappa distance quantifies the level of agreement between two sets and is defined as

$$\mathrm{Kappa}(A,B) = 1 - \frac{O-E}{1-E}$$

$$\text{with} \quad O = \frac{|A \cap B| + |(A \cup B)^c|}{|U|}$$
$$\text{and} \quad E = \frac{|A||B| + |A^c||B^c|}{|U|^2}$$

In this score, U is the union set of all genes [Yoon et al., 2019].

The Kappa distance score is symmetric (i.e., $\mathrm{Kappa}(A,B) = \mathrm{Kappa}(B,A)$) and the range of the resulting distances is $[0,2]$. A value of 0 indicates that the sets are completely identical, while a score of 2 reflects completely disjoint sets.

Since other distance metrics used in this work result in distances in the interval $[0,1]$, the Kappa distance score is transformed to match this range, ensuring comparability across all metrics. To normalise the Kappa distance score to the $[0,1]$ range, the following formula was applied:

$$\mathrm{Normalised\_Kappa}(A,B) = \frac{\mathrm{Kappa}(A,B) - min}{max - min}$$

where min and max represent the smallest and largest possible Kappa distances, respectively. Given that these values are defined as 0 and 2, the normalization simplifies to

$$\mathrm{Normalised\_Kappa}(A,B) = \frac{\mathrm{Kappa}(A,B)}{2}$$

This ensures all Kappa distances are scaled to the $[0,1]$ interval.

Due to its definition, the Kappa distance score adjusts for agreements which could occur by chance; a valuable advantage in its original scope of application. While this characteristic initially might not seem like an advantage in the field of gene set comparison, it can nevertheless be a valuable characteristic compared to the other scores presented in this thesis.

2.2.4 Protein-Protein Interaction-Weighted Meet-Min Distance Score

The Protein-Protein Interaction-Weighted Meet-Min (pMM) distance score is derived from the previously discussed Meet-Min distance score. Introduced in the work by authors Yoon et al. [2019], the pMM score incorporates protein-protein interaction data to enhance the MM score by weighting it with functional information at the protein level. While the MM score measures the overlap between gene sets, the pMM score adds a PPI-based factor, recognising that gene overlap alone may not fully capture functional

relationships between biological processes. Since functional interactions often occur at the protein level, the additional PPI component provides a more nuanced measure of similarity between gene sets.

The pMM distance score is defined as

$$\mathrm{pMM}(A, B) = min(\mathrm{pMMlocal}(A \to B), \mathrm{pMMlocal}(B \to A))$$

With

$$\mathrm{pMMlocal}(A \to B) = 1 - \frac{|A \cap B|}{min(|A|, |B|)} - \frac{\alpha}{min(|A|, |B|)} \sum_{a \in A - B} \frac{w \sum_{b \in A \cap B} P(a, b) + \sum_{b \in B - A} P(a, b)}{max(P)(w|A \cup B| + |B - A|)}$$

P is a PPI matrix, a numerical matrix indicating a possible interaction of two proteins through confidence scores in the range of $[0, 1]$. $P(a, b)$ is the interaction score of two proteins, represented by their coding genes a and b. α is a scaling factor to control the influence of the protein interactions on the resulting distances, with the range $[0, 1]$. If set to 0, the pMM score is identical to the MM score of Section 2.2.1, while a value of 1 balances the influence of the two components of the score. Lastly

$$w = \begin{cases} \dfrac{|A|}{|A| + |B|}, & \text{if } |A| \leq |B| \\ \dfrac{|B|}{|A| + |B|}, & \text{otherwise} \end{cases}$$

and $\mathrm{pMMlocal}(B \to A)$ is symmetrically defined.

The resulting pMM distance score of two gene sets A and B is symmetric (i.e., $\mathrm{pMM}(A, B) = \mathrm{pMM}(B, A)$) with a range of $[0, 1]$. A value of 0 indicates that the sets are either completely identical or have a high degree of functional similarity based on the provided interaction information, while a distance of 1 indicates completely disjoint sets.

The additional PPI component leverages the information inherent in interaction networks, making the score particularly suitable for analysing complex biological networks. It can reflect a more accurate representation of the functional relationships between gene sets.

2.2.5 Sørensen-Dice Distance Score

The Sørensen-Dice distance score is based on the equally named Sørensen-Dice coefficient, which was independently developed by both Sørensen and Dice in the 1940s. Therefore, it is also referenced as Sørensen-Dice index, Sørensen index and Dice's coefficient [Dice, 1945; Sørensen, 1948].

The distance score is defined as

$$\text{Sørensen-Dice}(A, B) = 1 - \frac{2|A \cap B|}{|A| + |B|}$$

The score is symmetric (i.e., $\text{Sørensen-Dice}(A, B) = \text{Sørensen-Dice}(B, A)$) and the range of the resulting distances is $[0, 1]$. A value of 0 indicates that the sets are completely identical, while a distance of 1 reflects completely disjoint sets.

The distance score measures the overlap between two sets with double the weight, making it particularly useful in scenarios where the overlap between sets is of higher importance than the overall number of elements in the sets. The Sørensen-Dice distance score can be applied in a variety of different scenarios such as ecology, computer lexicography and natural language processing [Connell, 1978; Zijdenbos et al., 1994; Rychlý, 2008].

2.2.6 GO Semantic Distance Score

The Gene Ontology is a bioinformatics database, which stores information on gene functions and interactions [Ashburner et al., 2000; Aleksander et al., 2023]. GO terms are frequently used in functional enrichment analyses to represent the individual gene sets and pathways in the results [Ashburner et al., 2000; Alexa et al., 2006; Yu et al., 2010; Wu et al., 2021; Aleksander et al., 2023]. As such, there are several different approaches and implementations to quantify the relationship and similarity between individual GO terms. Such GO semantic similarity scores can be used to assess the relationship between individual GO terms. One implementation can be found in the R package `GOSemSim` [Yu et al., 2010]. The package implements several semantic similarity measurements, divided into two method types: information content (IC) and graph-based methods.

The IC-based methods available in the package include the similarities by Resnik [Resnik, 1999], Lin [Lin, 1998], Schlicker [Schlicker et al., 2006] and Jiang [Jiang, Conrath, 1997]. These methods determine the similarity of two GO terms based on the information content of their closest common ancestor in the GO database, with some methods normalising by the information content of the two terms being compared. The individual IC of a GO term t is defined as the negative logarithmic probability of the term occurring in the GO database [Yu et al., 2010]:

$$IC(t) = -log(p(t))$$

The individual IC-based methods use different formulas to determine the semantic similarity between two GO terms. Resnik [Resnik, 1999] defines the similarity of two terms X and Y as

$$sim_{Resnik}(X, Y) = IC(MICA)$$

where the MICA is the most informative common ancestor of the two terms, i.e., the ancestor with the highest information content.

Lin [Lin, 1998] defines the similarity as

$$sim_{Lin}(X,Y) = \frac{2IC(MICA)}{IC(X)+IC(Y)}$$

The relevance method of Schlicker et al. [2006] combines both approaches of Resnik and Lin:

$$sim_{Rel}(X,Y) = \frac{2IC(MICA)(1-p(MICA))}{IC(X)+IC(Y)}$$

Lastly, Jiang's [Jiang, Conrath, 1997] method is defined as

$$sim_{Jiang}(X,Y) = 1 - min(1, IC(X)+IC(Y)-2IC(MICA))$$

The second type of GO similarity measures in the `GOSemSim` package are graph-based methods [Yu et al., 2010]. These methods leverage the topology of the underlying graph of the GO database, which is a directed acyclic graph (DAG) where nodes represent GO terms and edges represent hierarchical relationships such as "is-a" or "part-of" [Ashburner et al., 2000; Yu et al., 2010; Aleksander et al., 2023]. The graph-based method implemented in the package is the one proposed by Wang et al. [2007]. For the graph-based method of Wang, a GO term X is formally represented as

$$DAG_X = (X, T_X, E_X)$$

where T_X is the set containing the term X and all of its ancestors and E_X is the set of edges connecting the GO terms in DAG_X. Then, the method of Wang et al. [2007] for two terms X and Y is defined as

$$sim_{Wang}(X,Y) = \frac{\sum_{t \in T_X \cap T_Y} S_X(t) + S_Y(t)}{SV(X)+SV(Y)}$$

$$\text{with} \quad SV(X) = \sum_{t \in T_X} S_X(t)$$

$$\text{and} \quad S_X(t) = \begin{cases} S_X(X) = 1, \\ S_X(t) = max(w_e S_X(t') | t' \in \text{children of } t), & \text{if } t \neq X \end{cases}$$

where w_e is the semantic contribution factor, i.e., weight for edge $e \in E_X$, which links term t with its child term t'.

All of the discussed IC and graph-based scores increase with similarity rather than distance, i.e., a score of 1 represents identical gene sets. Hence, in order for the semantic similarity scores to be comparable to the other distance metrics, the similarity score is inverted into a distance score by subtracting the calculated similarity score from 1. The resulting distance measure is symmetric across all the methods presented. While this is fairly evident for the IC-based approaches, it may not be as intuitive from Wang's

method's description and formula. Despite the GO database being structured as a DAG, Wang's method ensures symmetric similarity scores by equally considering both GO terms' hierarchical relationships and shared ancestors, ensuring the direction of comparison does not affect the outcome [Wang et al., 2007; Yu et al., 2010].

All scores besides Resnik and Jiang range from $[0, 1]$ with similar interpretations compared to the other available metrics. Therefore, a normalization is applied to the score of Resnik and Jiang before transforming the similarity score to a distance score, following the same approach already discussed for the Kappa distance score. The applied normalization is defined as

$$\text{Normalised\_GOSimilarity(X,Y)} = \frac{\text{GOSimilarity}(X, Y) - min}{max - min}$$

where min and max are the smallest and largest similarity scores, respectively, and $\text{GOSimilarity}(X, Y)$ is the similarity score of two terms X and Y calculated with either Resnik's or Jiang's method.

Compared to the other distance scores, the GO semantic distance score is only applicable to data where gene sets and pathways are represented by GO terms. While this might limit the overall applicability of the distance score, the score leverages the specific characteristics of the GO database making it especially suitable in cases where these specific terms are of interest, e.g., when the focus is more on gene functions rather than direct gene overlap.

2.3 Definition of the Clustering Algorithms

The introduced distance scores and their implemented visualisations can be used to observe some initial, prominent patterns in the data, offering rudimentary aggregation information as visual patterns without further composition or structural information. Therefore, in order to amplify this additional pattern information, a subsequent processing step is applied in the form of clustering. Clustering algorithms are used to organise gene sets into coherent groups, which can be either distinct (non-overlapping) or shared (overlapping), allowing researchers to focus on functionally related clusters rather than individual gene sets.

As individual research situations require specific properties of clustering algorithms, and each clustering algorithm offers different points of view about the data in question, four distinct methods have been selected and incorporated into the presented version of `GeDi`. These clustering algorithms are:

- Louvain Clustering Algorithm
- Markov Clustering Algorithm
- Fuzzy Clustering Algorithm
- Partitioning around Medoids Clustering Algorithm

2.3.1 Louvain Clustering Algorithm

The Louvain clustering algorithm is a widely used method for detecting communities in large data networks [Blondel et al., 2008]. The algorithm optimises a modularity score which quantifies the strength of division of a network into clusters[3]. The score measures how strongly nodes within a cluster are interconnected compared to the connections between nodes in different clusters [Blondel et al., 2008]. The modularity Q of a graph can be defined as

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} is the weight of the edge between nodes i and j. k_i and k_j are the sums of the weights of the edges connected to nodes i and j respectively. c_i and c_j are the clusters to which nodes i and j belong. Lastly, m is the total weight of all edges in the graph and

$$\delta(c_i, c_j) = \begin{cases} 1, & \text{if } c_i \text{ and } c_j \text{ are the same cluster} \\ 0, & \text{otherwise} \end{cases}$$

The Louvain algorithm starts with an initial setup in which each node in the network is treated as an individual cluster [Blondel et al., 2008]. In the context of `GeDi`, each node represents a gene set in the initial layout and edges connect gene sets with a pairwise distance below a user-defined threshold. The edge weights are based on the distances between gene sets. Since the Louvain algorithm optimises the modularity score, the distances are converted to similarity values to ensure that the algorithm forms clusters of closely related gene sets. This is achieved by subtracting the distances from 1.

In the first phase, the algorithm iteratively reassigns nodes to different clusters to achieve a new configuration which maximises the modularity score Q. This process is continued until no further improvement in the modularity score of the graph can be achieved.

Once the modularity score of the current configuration is maximised, the Louvain algorithm proceeds to the second phase, the aggregation phase. During the aggregation phase, each cluster is collapsed into a single, representative node and the edges between the clusters are summed based on the connections between member nodes. Afterwards, the process of iterative reassignment and aggregation is repeated until no further improvement of the modularity score can be achieved [Blondel et al., 2008].

The Louvain clustering algorithm is especially useful for the detection of clusters in large and complex networks. Particularly, it is well-suited for networks in which clusters are densely interconnected with sparse connections in between clusters, such as in scRNA data [Seth et al., 2022]. However, this makes the algorithm also prone to favour large clusters over smaller ones within the network, as it seeks to maximise the modularity score. Additionally, the Louvain algorithm is non-deterministic, meaning the output can vary between individual executions, making it harder to reproduce results consistently.

[3]The terms cluster and community both refer to groups of nodes that are more densely connected to each other than to those outside the group. These terms will be used interchangeably throughout Section 2.3

Lastly, it should be mentioned that the Louvain algorithm results in non-overlapping clusters. While non-overlapping clusters might not accurately represent biological networks in which genes and gene groups can be part of multiple functional groups or pathways, there are certain research situations where the generation of non-overlapping clusters is favourable. Examples of such could be the investigation of developmental stages, in which a clear separation of early and late stages could be intended.

2.3.2 Markov Clustering Algorithm

The Markov clustering algorithm is a graph-based clustering method which identifies clusters by simulating random walks within the network. The algorithm leverages the idea that random walks tend to stay longer within densely connected regions of the graph (i.e., clusters), and transition less frequently to sparsely connected regions [van Dongen, 2000].

The algorithm starts with an initial graph in which each node represents an individual cluster and edges represent interactions or similarities between nodes. In the case of `GeDi`, nodes represent individual gene sets and weighted edges are formed between them. The weight of each edge is determined by the calculated distance between gene sets, whereas distances and edges above a user defined threshold are omitted. This is used as a means to reduce the runtime of the algorithm. In order to ensure the formation of closely related clusters, the distances are converted to similarity scores as already described for the Louvain algorithm.

In the first step of the Markov clustering, a stochastic transition matrix is created where each element of the matrix represents the transition probability from one node to another node during a random walk. The transition matrix M is derived from the adjacency matrix A of the graph, where each element M_{ij} represents the probability of moving from node i to node j in a single step of a random walk defined as [van Dongen, 2000]

$$M_{ij} = \frac{A_{ij}}{\sum_k A_{ik}}$$

where A_{ij} represents the edge weight between nodes i and j in the adjacency matrix and k sums over all nodes connected to i. The transition matrix is a normalised version of the adjacency matrix, ensuring that the sum of each column equals 1. This transition matrix forms the basis for simulating the movement of random walks across the network.

In the following expansion step, the algorithm proceeds to mimic random walks across the graph. During the expansion step, the transition matrix M is first raised to a power p, resulting in the expanded transition matrix M_{exp} (i.e., $M_{exp} = M^p$). This power is typically set to 2 for reasons of balancing local and global structures as well as efficiency of the computation. By raising the transition matrix to a power, the algorithm distributes probabilities across extended paths in the graph, taking into account both direct links and connections that span multiple intermediate nodes. This helps reveal relationships between nodes that are not immediately adjacent and enhances the detection of clusters by effectively expanding the influence of each node within its local neighbourhood [van Dongen, 2000].

Following the expansion step, the algorithm applies the inflation step. In this step, each element of the expanded matrix M_{exp} is raised to a power greater than 1. This inflation parameter, typically denoted by r, amplifies higher transition probabilities and diminishes lower ones. The default value for r is usually also set to 2, enhancing cluster separation while simultaneously controlling for cluster granularity. The value for two nodes i and j in the inflation matrix M_{infl} can be calculated as [van Dongen, 2000]

$$M_{infl}(i,j) = \frac{M_{exp}(i,j)^r}{\sum_k M_{exp}(i,k)^r}$$

The inflation step reinforces the cluster boundaries, ensuring that nodes within a cluster remain more closely associated, while connections between different clusters become weaker.

The expansion and inflation step are iteratively repeated until the matrix converges. In the Markov clustering algorithm, this means that the matrix no longer significantly changes and the clusters remain unchanged. At convergence, the network reveals distinct clusters, where nodes within the same cluster exhibit strong internal connections, while connections between clusters are sparse [van Dongen, 2000].

The Markov clustering algorithm is suitable for a variety of network data with varying degrees of complexity. Especially the inflation step gives the algorithm the flexibility to detect clusters of various sizes. The inflation parameter r negatively correlates with the cluster size, meaning higher values of r lead to smaller, tighter clusters, while smaller values lead to larger clusters [van Dongen, 2000]. However, this also means that the results are highly influenced by the choice of inflation parameter, hence requiring a careful adjustment to achieve meaningful clusters for a particular dataset.

2.3.3 Fuzzy Clustering Algorithm

The Fuzzy clustering algorithm, as implemented in the DAVID database, is a method used to cluster gene sets based on their functional similarity [Huang et al., 2007, 2009]. Unlike the previously discussed clustering algorithms, nodes (i.e., gene sets) can belong to multiple clusters.

The algorithm begins by selecting initial cluster seeds, which are groups of gene sets that share a strong functional relationship. Gene sets are considered to have a strong functional relationship, if their distance score is smaller or equal than a specified similarity threshold. The selection of these initial seeds can be further refined by applying a membership threshold, which specifies the minimum number of genes that must be strongly functionally related for the group to qualify as a seed [Huang et al., 2007, 2009]. The membership threshold ensures that only groups with a strong functional core are considered for clustering.

In an iterative second step, the initial seeds are merged into larger groups until the final clusters are determined. Seeds sharing a significant number of common members are merged, forming larger, more comprehensive clusters. This merging process is controlled by a clustering threshold, which defines the minimum amount of shared members required for two clusters to be merged. The iterative nature of this step allows the al-

gorithm to refine the clusters progressively, ensuring that gene sets with overlapping functions are grouped together in a meaningful way [Huang et al., 2007, 2009]. A compelling and easy to understand example walk-through of the Fuzzy clustering algorithm can be found in Supplementary File 13 of the original publication by Huang et al. [2007].

A key feature of the Fuzzy clustering algorithm is the possibility of nodes to be part of several clusters. Especially in the context of biological data and gene sets, this approach reflects the characteristic that gene set functions and members can be overlapping as biological processes are not usually disjoint. Therefore, by enabling gene sets to belong to multiple clusters, the algorithm provides a more nuanced representation of gene-function relationships and provides insights which might be missed by more rigid clustering methods [Huang et al., 2007, 2009]. However, this usually leads to more complex results, as cluster boundaries might not be well defined. This can lead to redundancy in the results especially if two clusters share an amount of gene sets, which is not large enough to allow for merging but still substantial enough to hinder the interpretation.

2.3.4 Partitioning around Medoids Clustering Algorithm

The Partitioning around Medoids (PAM) clustering algorithm is a partitioning method widely used in the fields of market research and healthcare analysis [Mushtaq et al., 2018; Alie, Gustriansyah, 2024]. The PAM algorithm is very similar to the more commonly known k-Means algorithm, with both methods designed to divide a dataset into k distinct clusters [Lloyd, 1982; Kaufman, Rousseeuw, 1987]. But unlike the centroids used in k-Means, the PAM algorithm represents each cluster by a medoid, which is the most centrally located actual data point within each cluster. The PAM algorithm aims to minimise the distance between data points and their assigned medoids, leading to compact and well-separated clusters [Kaufman, Rousseeuw, 1987].

The algorithm starts with k initial medoids, whereas k is a user-defined number. These initial medoids are chosen from the available data points (i.e., gene sets) during the build phase of the algorithm. These initial medoids are chosen to minimise the overall dissimilarity to all other gene sets, ensuring that they serve as central, representative points for each cluster from the onset [Kaufman, Rousseeuw, 1987].

In the next step of the algorithm, gene sets are assigned to the nearest medoid based on previously calculated distances. After all gene sets have been assigned to clusters, the algorithm evaluates whether any non-medoid gene set would serve better as medoid. For this step, each medoid is swapped with a non-medoid and the total dissimilarity of the graph is calculated. Here, the total dissimilarity is the sum of the dissimilarities between all gene sets and their closest medoid across all clusters. If the swap improves the overall clustering by reducing the total dissimilarity, the new medoid is accepted and all gene sets are again assigned to their nearest medoid. This process of medoid swapping and cluster reassignment is repeated iteratively [Kaufman, Rousseeuw, 1987; Maechler et al., 2023].

The algorithm continues until no medoid swap can reduce the total dissimilarity in the graph [Kaufman, Rousseeuw, 1987; Maechler et al., 2023]. The goal of the algorithm is to minimise the total dissimilarity between data points and their nearest medoids across

all clusters, which ensures that the gene sets are assigned to meaningful and functionally similar clusters.

The PAM algorithm is widely applied across various fields due to its ability to handle outliers effectively, as it selects medoids from the actual data points. Its versatility is further enhanced by the ability to work with different distance metrics or precomputed distances. The parameter k determines the number of clusters, influencing the detail level of the results: higher values of k yield smaller, more refined clusters, while lower values produce larger, broader clusters, thus affecting the overall granularity and interpretability of the results.

Nonetheless, it should be noted, that the algorithm can be computationally expensive, especially for large datasets, as evaluating all possible swaps of medoids requires a considerable amount of time. This challenge has been addressed through optimised implementations, such as those available in the R package `cluster` [Maechler et al., 2023], which offers different, slightly varied implementations of the algorithm to optimise the runtime. Readers are encouraged to consult the original documentation of the `cluster` package [Maechler et al., 2023] for a detailed description of the differences between the available options.

2.4 Implementation and Features of the `GeDi` Package

In an effort to facilitate and streamline the interpretation of functional enrichment analysis results, `GeDi` was developed as an R/Bioconductor package. The package implements a wide array of functionality, including the described distance metrics and clustering algorithms, to aggregate functional enrichment results into groups of similar gene sets, thereby easing interpretation and hypothesis generation. Additionally, `GeDi` also incorporates network information into the analysis, e.g., by using protein-protein interaction information.

`GeDi` is also embedded in the R/Bioconductor package ecosystem, allowing for the use of various different functional enrichment approaches implemented in different R packages, such as `topGO` and `clusterProfiler` [Alexa et al., 2006; Wu et al., 2021]. This integration makes `GeDi` a well-suited solution for incorporation into various analysis workflows across various data types, such as bulk and single-cell RNA-seq, proteomics, and spatial transcriptomics. It summaries the potentially overwhelming amount of information contained in the results of an enrichment analysis, allowing for a more streamlined interpretation and generation of new hypotheses. The `GeDi` package is available under the MIT license on Bioconductor (`https://bioconductor.org/packages/GeDi`), with the development version available on GitHub (`https://github.com/AnnekathrinSilvia/GeDi`).

A key aspect of `GeDi`'s development was to consolidate diverse functionalities within a unified tool, leveraging existing, well-established methods rather than re-implementing them. This design choice allowed the integration of various tools and methods into one package, while ensuring that `GeDi` serves as an accessible, multifunctional platform for data exploration and interpretation, without reinventing already established approaches.

To ensure compatibility with various implementations of functional enrichment analysis and to accommodate a diverse user base, GeDi accepts the input in various formats such as text files, spreadsheets or in the form of standardised containers, which are often used in the R community. The input data must consist of at least two columns: one containing gene set identifiers and another containing gene identifiers of the genes in each gene set. These columns should be labelled as "Genesets" and "Genes", respectively.

There are no strict requirements for the identifier (ID) types used for either the gene sets or genes. However, GeDi strongly encourages the use of standardised and widely known identifiers, such as HUGO Gene Nomenclature Committee (HGNC) gene symbols [Seal et al., 2023], ENSEMBL [Yates et al., 2020], or GENCODE [Frankish et al., 2019] for genes, and pathway identifiers from known databases such as GO [Ashburner et al., 2000; Carbon et al., 2019], KEGG [Kanehisa et al., 2017, 2019], the Reactome Pathway Database [Fabregat et al., 2018], or the MSigDB [Liberzon et al., 2011, 2015] for gene sets in order to enhance reproducibility.

In the following sections, the methods and functions available for data processing in GeDi will be discussed. First, Section 2.4.1 will discuss available data preparation and preprocessing functions. Afterwards, Section 2.4.2 and Section 2.4.3 will discuss the implementation of the individual distance metrics and their corresponding visualisations, while Section 2.4.4 and Section 2.4.5 will discuss the respective implementation of the clustering algorithms and the corresponding visualisations. All functionality discussed in these sections and their implementation can be found in the GitHub repository of the package (`https://github.com/AnnekathrinSilvia/GeDi`). Readers of this thesis are encouraged to refer to this repository for the detailed implementation and documentation of all downstream discussed functionality.

2.4.1 Data Preparation and Preprocessing

In order to support the various data formats for the input data, a data preparation is needed in GeDi. This can be followed by two optional data preprocessing steps, which are gene set filtering and the download of PPI information. Depending on the input data as well as the research question, these steps can improve the exploration and interpretation of functional enrichment results.

2.4.1.1 Data Preparation of the Input Data

The necessary data preparation in GeDi is performed by the **prepareGenesetData()** function, which is designed to process and extract the gene information from the given data. Functional enrichment implementations, such as those provided in the R packages topGO [Alexa et al., 2006] and clusterProfiler [Wu et al., 2021], store the gene information in compact formats within a column of the results (e.g., as a single string separated by delimiters like commas or semicolons). This function splits those gene strings into individual gene lists, a format which is easier to use in downstream analysis steps. Additionally, the function automatically detects the separator used within the gene strings to further automate the data preparation and reduce the likelihood of errors.

The **prepareGenesetData()** function has two arguments: **genesets**, representing the input data and **gene_name**, specifying the column containing the strings of gene

identifiers. If the latter argument is not provided, the function will assume that this column is called "Genes". Otherwise, the **gene_name** argument can be used to specify an alternative column name. This is a further design choice to ensure compatibility with a large variety of different implementations of functional enrichment analyses.

2.4.1.2 Optimizing Analysis through Gene Set Filtering

The first optional preprocessing step involves the filtering of gene sets from the data. Oftentimes, functional enrichment results include large, generic or even redundant gene sets among the extensive list of results. These broad terms are associated with a large number of genes, hence increasing their likelihood to include differentially expressed genes and to appear among the top affected gene sets in a functional enrichment analysis.

This redundancy and overlap between gene sets can mask processes involving smaller and more specific gene sets, which appear much later in the ranked list of results, hence hindering the interpretation of overall interactions in the dataset. Examples for such generic terms include "biological process", "cellular process", "metabolic function" or "cellular component" in the GO database or "Metabolism", "Cellular Processes" or "Human Diseases" in the KEGG database [Ashburner et al., 2000; Kanehisa et al., 2017; Carbon et al., 2019; Kanehisa et al., 2019].

Moreover, these large gene sets can considerably increase the overall runtime of the analysis without adding substantial information. While these gene sets can still be highly valuable during data interpretation, removal of these gene sets should be considered when interpreting and exploring the results. `GeDi` provides the option to filter these gene sets from the data using the **`filterGenesets()`** function.

To assist users in identifying which gene sets to filter, the **`gsHistogram()`** function is implemented in `GeDi`. The function plots a histogram of the size (i.e., number of genes) of each gene set in the input data. A histogram is a graphical representation used to show the frequency distribution of values in a dataset (Figure 5). It displays data by grouping numerical values into bins and representing the frequency of data points within each bin using bars. The height of each bar reflects how many data points fall within that particular range. In the **`gsHistogram()`** function of `GeDi`, the size distribution of each gene set is plotted based on the input functional enrichment data. Besides the input data, the function also has arguments to control the size of each bin as well as arguments to restrict the shown histogram to a specific range. Per default, the range will be defined by the smallest and largest available gene set.

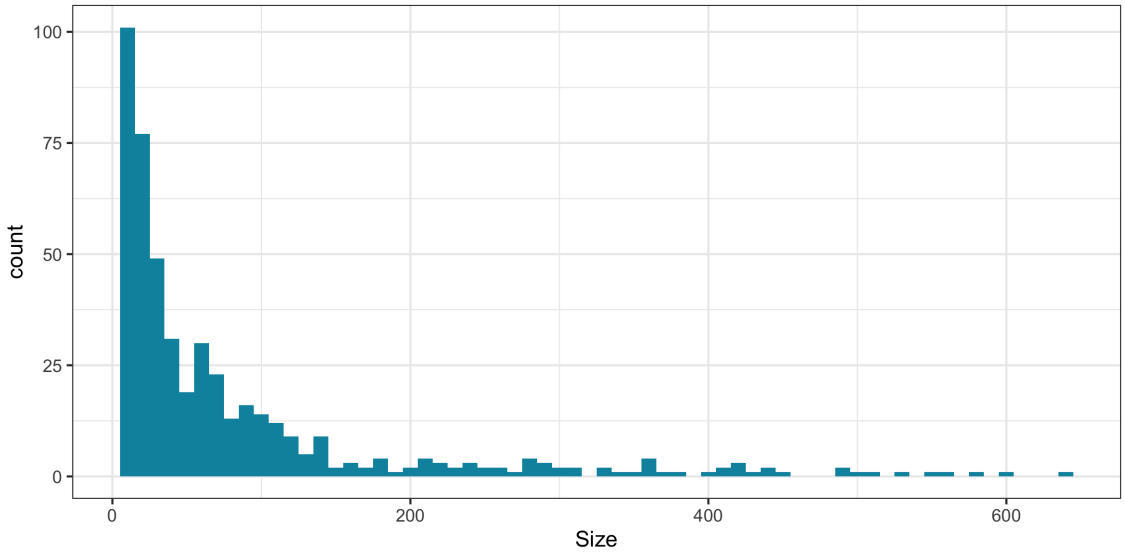


Figure 5 – Histogram of Gene Set Sizes
The figure shows an example of a histogram plotted with `GeDi`, which visualises the distribution of the individual gene set sizes of the data. The abscissa indicates the size of each gene set (i.e., number of genes), while the ordinate shows the frequency of each size. The chosen bin size was 5.

2.4.1.3 Download of Protein-Protein Interaction Data

In order to support the network component of the pMM distance score, species-specific protein-protein interaction data is needed. `GeDi` includes the **`getPPI()`** function to download this PPI data. This function requires a gene list, ideally obtained through the **`prepareGenesetData()`** function, as well as a `STRINGdb` object from the equally named package [Szklarczyk et al., 2023], and an annotation object that maps STRING IDs to gene names.

This function will fetch the PPI data from the STRING (Search Tool for Recurring Instances of Neighbouring Genes) database, a biological database providing information on known and predicted protein-protein interactions [Szklarczyk et al., 2023]. The `STRINGdb` object can also be generated using **`getStringDB()`**, which will download the PPI information for the species of the input data. Additionally, this function can cache the downloaded data, enabling faster access in future analyses by reusing the cached information. Caching is particularly beneficial when repeatedly analysing data of the same species, as it eliminates the need to download the PPI data multiple times. Moreover, `GeDi` includes a **`getAnnotation()`** function to create the annotation object, which is used in the **`getPPI()`** function to map gene names to STRING IDs.

2.4.2 Distance Scoring within `GeDi`

`GeDi` uses various implemented distance metrics (as described in Section 2.2) to quantify the (dis)similarity between individual gene sets. These resulting distances are used to aggregate the initially large and extensive results of a functional enrichment analysis, thereby facilitating the interpretation of the data.

The six available distance metrics are implemented in the following functions:

- **getMeetMinMatrix(genesets, progress = NULL, BPPARAM = BiocParallel::SerialParam())**
- **getJaccardMatrix(genesets, progress = NULL, BPPARAM = BiocParallel::SerialParam())**
- **getKappaMatrix(genesets, progress = NULL, BPPARAM = BiocParallel::SerialParam())**
- **getpMMMatrix(genesets, ppi, alpha = 1, progress = NULL, BPPARAM = BiocParallel::SerialParam())**
- **getSorensenDiceMatrix(genesets, progress = NULL, BPPARAM = BiocParallel::SerialParam())**
- **getGODistanceMatrix(genesets, method = "Wang", ontology = "BP", species = "org.Hs.eg.db", progress = NULL, BPPARAM = BiocParallel::SerialParam())**

The functions have similar arguments, with the pMM and GO semantic similarity calculation having additional, method-specific arguments. All scoring functions take the **genesets** as input data, which should be the lists of genes as returned by the previously discussed **prepareGenesetData()** function. Additionally, each function returns the output as a `Matrix` object from the package by Bates et al. [2024], containing the pairwise calculated distances for each pair of gene sets.

To calculate the pMM distances using the **getpMMMatrix()** function, two additional arguments are needed. As discussed in Section 2.2.4, this specific score uses protein-protein interaction information to quantify the (dis)similarity of gene sets. This PPI information should be provided to the function in the form of a `data.frame` object via the **ppi** argument. This `data.frame` object should contain three columns, two columns called "Gene1" and "Gene2" containing gene identifiers and a third column called "combined_score" containing a numerical confidence score for the respective interaction. The PPI information can be downloaded from the STRING database using the previously discussed **getPPI()** function. The influence of the provided PPI information on the resulting distances can be regulated using the **alpha** argument of the function.

The **getGODistanceMatrix()** function arguments follow the original implementation in the `GOSemSim` package of Yu et al. [2010]. The function argument **method** specifies the method of GO semantic similarity to use. Possible options are the methods of Resnik, Lin, Rel, Jiang and Wang, which were already discussed in Section 2.2.6. The **ontology** argument refers to the GO ontology which should be used, with the options being "BP" (Biological Process), "MF" (Molecular Function) and "CC" (Cellular Component) [Yu et al., 2010]. Lastly, the **species** argument defines the species of the input data and should follow the naming conventions of the annotation packages available in Bioconductor (e.g., "org.Hs.eg.db" for Homo Sapiens) [Huber et al., 2015]. The available annotation packages can be found under the following link: `https://www.bioconductor.org/packages/release/data/annotation/`.

All available scoring functions include the **progress** argument, which must be an object of the type `shiny::Progress()` [Chang et al., 2024]. This object is primarily used in the interactive `Shiny` application of `GeDi`, which will be discussed in Section 2.5. In the application, the progress object controls a small progress bar displayed in the lower right corner of each panel (Figure 6). The progress bar is a helpful visual indicator of the internal computation of the application, particularly useful for keeping users informed during long-running computations. Progress bars are consistently used throughout `GeDi` to update users on ongoing tasks. When scoring functions are used independently, where a progress bar may not be necessary, the `progress` parameter can be set to NULL. This is also the default behaviour of the functions, implemented to prevent progress bar updates in the stand-alone execution of the functions, as `shiny::Progress()` objects are primarily designed to be used within `Shiny` applications.

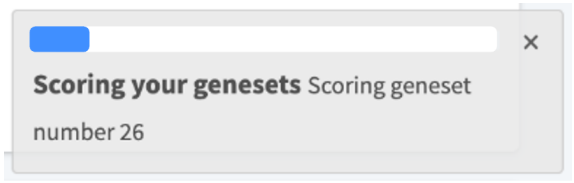


Figure 6 – Progress Bar of Gene Set Distances Calculation
The figure shows an example of the progress bar, which will be displayed in several steps in the `GeDi` application. This progress bar informs users on the current progress of specific, lengthy computations. The specific example shows the progress bar during the calculation of gene set distances.

The last argument shared by all scoring functions is the **BPPARAM** argument (standing for `BiocParallelParam`). As functional enrichment analysis results can contain a substantial amount of gene sets, the computation of pairwise gene set distances can be computationally resource and time-consuming. Therefore, all scoring functions are implemented to support concurrent execution. This is achieved using the `bplapply` function from the `BiocParallel` package [Morgan et al., 2024]. The `BiocParallel` package provides infrastructure for parallel computing in R, thus reducing the runtime of calculations through the use of multiple cores. This allows independent tasks to run concurrently on separate cores, making analyses more efficient. The `bplapply` function is a parallelized version of the commonly used `lapply` function; an R base function which will apply a specified function to each element of a list. In each of the distance scoring functions, the **BPPARAM** argument specifies which parallel backend will be used during execution. The default choice in `GeDi` is the **`SerialParam()`**. While this defaults to a serial backend with sequential execution, this choice ensures compatibility across all operating systems, including Windows, where some `BiocParallel` backends are unavailable [Morgan et al., 2024]. Nevertheless, the design maintains the flexibility to scale up by switching to a parallel backend when required, offering users an adaptable solution for more intensive computations.

2.4.3 Visualisations of the Gene Set Distances

While distance metrics can be used to quantify the (dis)similarity between individual gene sets, they themselves can be challenging to interpret especially since there would be $\binom{n}{2} = \frac{n(n-1)}{2}$ different combinations for n input gene sets. To address this, `GeDi` offers various visualisations for the calculated distances to aid in their interpretation. These

visualisations include a heatmap, a dendrogram, and a network representation of the distances, making it easier to understand the relationships between gene sets.

2.4.3.1 Heatmap Visualisation

Heatmaps are graphical representations of data values as colours in a matrix format [Gu et al., 2016; Gu, 2022]. Heatmaps are commonly used to visualise patterns, correlations or distributions in large datasets. They are particularly useful for analysing high-dimensional datasets, as they provide a clear and intuitive visualisation of the relationships between multiple variables. In the field of bioinformatics, heatmaps are often employed to display gene expression levels across different conditions or samples, where the intensity of the colour corresponds to the magnitude of expression [Gu et al., 2016; Gu, 2022]. In `GeDi`, heatmaps are used to visualise the distance score results between individual gene sets, where each cell in the matrix corresponds to the distance between two gene sets.

Figure 7 shows an example of a heatmap generated with **`distanceHeatmap()`** function of `GeDi`. This function takes a matrix of distances between gene sets and plots a heatmap of these scores. Rows and columns are labelled by the gene set identifier, which can also be omitted to enhance clarity of the visualisation as shown in Figure 7. In the resulting heatmap, small distances are visualised in blue, while large distances are plotted in red; a colour palette chosen for its accessibility to people with various types of colour blindness. The heatmap allows users to quickly assess the relationships between gene sets and identify initial clusters of similar gene sets or distinct outliers. For the actual generation of the heatmap, `GeDi` uses the functionality provided in the `ComplexHeatmap` package [Gu et al., 2016; Gu, 2022], a widely used Bioconductor package to visualise various types of data in a heatmap.

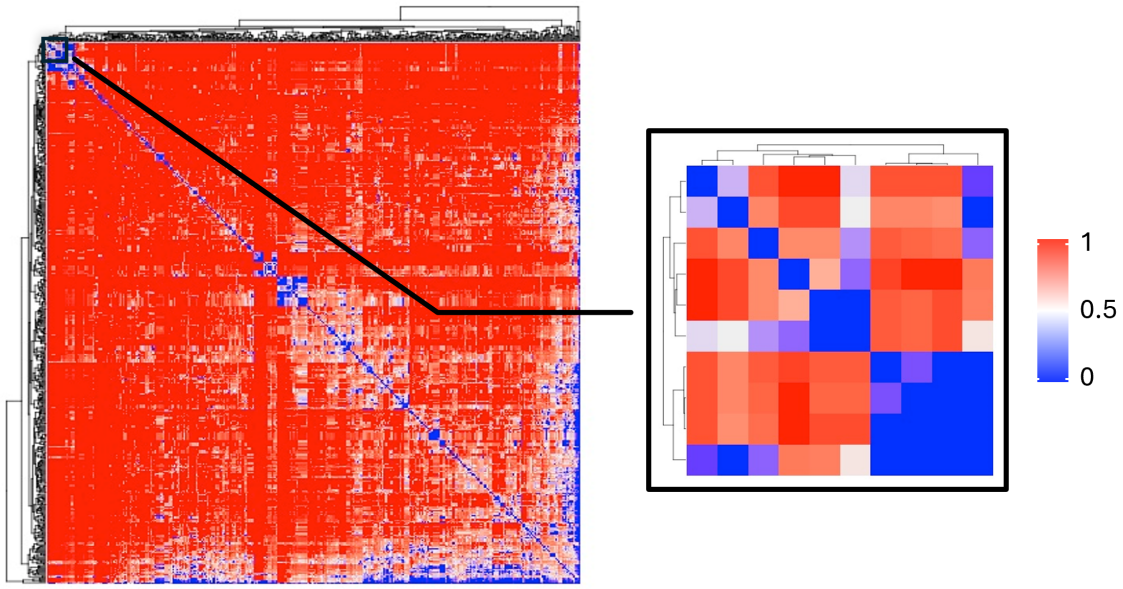


Figure 7 – Heatmap of Gene Set Distances
The figure shows an example of a heatmap of gene set distances generated with `GeDi`. On the left side, the heatmap plotted with the **`distanceHeatmap()`** function can be seen, with a zoomed-in section of the upper left corner shown on the right. The colour represents the distance between pairs of gene sets, with darker blue indicating a small distance and red indicating a large distance. For sake of simplicity, row and column labels have been excluded in the shown version.

2.4.3.2 Dendrogram Visualisation

The second visualisation type is a dendrogram. Dendrograms are tree-like diagrams which visualise the arrangement of clusters formed through hierarchical clustering [Gallery, 2024a]. Usually applied in evolutionary studies, dendrograms can also be used to demonstrate the relationships between genes and gene sets [Paradis et al., 2004; Gamermann et al., 2019]. In the context of `GeDi`, dendrograms are used to represent the relationships between gene sets based on their distances. Each branch in the dendrogram corresponds to a gene set, and the length of the branches reflects the degree of dissimilarity between them. Gene sets that are more similar are grouped together at lower branch points, while those that are more distant are connected higher up in the tree [Gallery, 2024a; de Vries, Ripley, 2024].

Figure 8 shows an example of a dendrogram generated with the **`distanceDendro()`** function of `GeDi`. The function takes a matrix of gene set distances and creates a dendrogram to visually represent the hierarchical clustering of the gene sets. The plot is based on the functionality implemented in the ggdendro package [de Vries, Ripley, 2024], while the hierarchical clustering is implemented using the `hclust` function of the `stats` package [R Core Team, 2024]. The agglomeration method, i.e., the hierarchical clustering approach used to decide at which point gene set branches should be connected, can also be specified in the **`distanceDendro()`** function. The available options are identical to the ones presented by the `hclust` functions in the `stats` package [R Core Team, 2024]. Readers are referred to the original documentation of the `stats` package for a more detailed description of the available options and their differences.

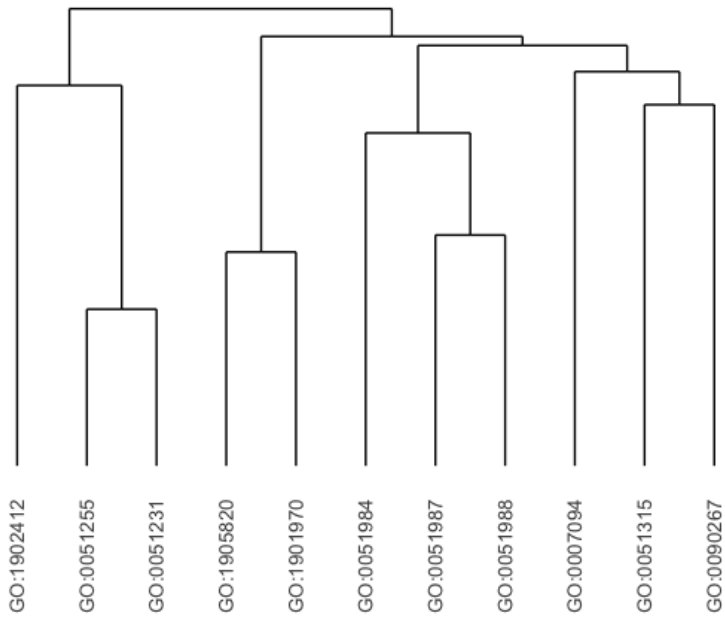


Figure 8 – Dendrogram of Gene Set Distances
The figure shows an example of a dendrogram of gene sets based on their calculated distances. Each branch represents a gene set, and the height at which branches merge indicates the degree of similarity between the sets. Gene sets with shorter branch lengths are more closely related, while those merging at higher points show greater dissimilarity.

2.4.3.3 Network Visualisation

The last available visualisation of the distances is a network-like representation. Compared to the matrix-based visualisations of the heatmap and the dendrogram, the network represents the calculated distances as a graph, in which each gene set is a node and the relationships between gene sets are represented as edges. Edges are drawn between gene sets which are similar (i.e., with a distance below a certain cutoff threshold), forming clusters of related gene sets within the network. This visualisation allows for a more dynamic exploration of the distances and can highlight connections which might not immediately be visible in the previously discussed visualisations.

An example for a network visualisation of the distances can be seen in Figure 9. In `GeDi`, the **`buildGraph()`** function constructs an undirected graph from an adjacency matrix. The adjacency matrix is derived from the gene set distances and indicates which gene sets (nodes) are connected by edges based on a cutoff value. This is done by the **`getAdjacencyMatrix()`** function, which takes a matrix of distances and a numeric cutoff value as arguments. Based on this information, the **`buildGraph()`** function builds the network-like representation of the gene set distances. The function additionally has arguments for the input gene set data and the gene set identifiers; information which is used to generate titles and additional information for the individual nodes in the graph. These titles as well as the additional information, which is available upon hovering over a node, are especially useful in the interactive version of this graph provided in the `Shiny` web application of `GeDi`, discussed in Section 2.5. The graph visualisation of `GeDi` is strongly based on the functions available in the `igraph` R package [Csárdi et al., 2024], a commonly used package to create and analyse graphs and network structures.

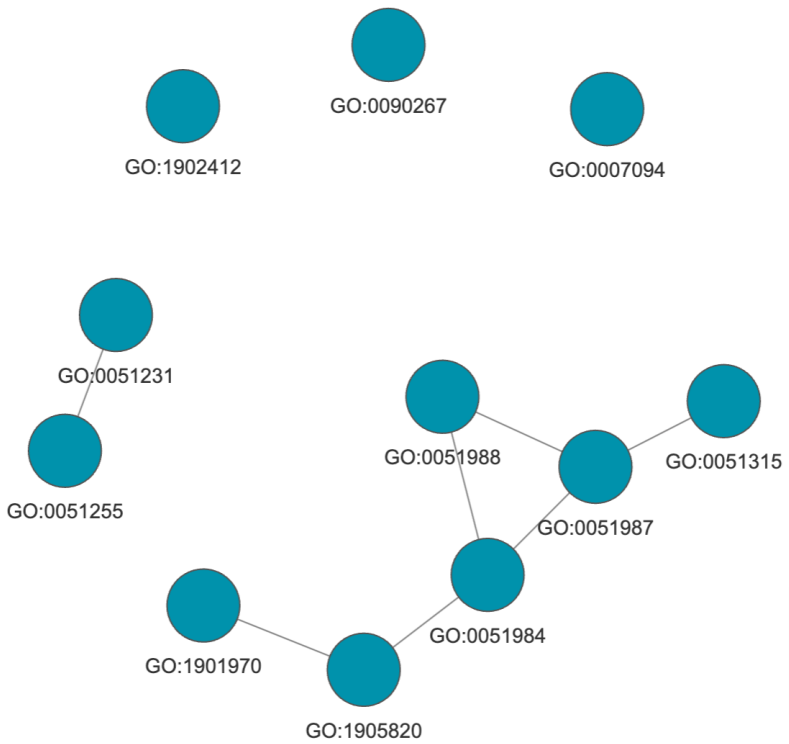


Figure 9 – Network of Gene Set Distances
The figure shows an example of a network visualisation of the gene set distances. Each node represents a gene set and edges connect gene sets whose distance is below a specified threshold.

2.4.4 Clustering within GeDi

In order to further streamline and facilitate the interpretation of functional enrichment results, GeDi implements various clustering algorithms to group gene sets based on their calculated distances. These clustering methods help refine the functional enrichment results by organising (functionally) related gene sets into coherent clusters, further simplifying data interpretation.

The four available clustering algorithms, as outlined in Section 2.3, are implemented in the following functions:

- **louvainClustering(scores, threshold)**
- **markovClustering(scores, threshold)**
- **fuzzyClustering(seeds, threshold)**
- **pamClustering(scores, k)**

During the development of GeDi, it became evident that both the Louvain and Markov clustering algorithm rely on the identical input data as well as pre- and postprocessing steps, such as the generation of the initial graph for the clustering or the postprocessing of the results for further use in GeDi. Hence, in order to optimise efficiency and avoid redundancy, I decided to implement both clustering algorithms within a single function, which is called **clustering()**. This function uses an argument called **cluster_method** to decide which algorithm to apply to the data. This design reduces the amount of duplicated code in the code base of GeDi, minimising the risk of error propagation through copying and pasting nearly identical code chunks. In the same manner, future maintenance is simplified as updates and bug fixes need to be applied only once. While this design has clear advantages, I recognised that some users might expect separate, intuitively named functions for each clustering algorithm, which could lead to confusion when incorporating the package into their workflows. To address this issue, I also implemented the **louvainClustering()** and **markovClustering()** functions to serve as wrapper functions around the main **clustering()** function. In the end, this implementation approach ensures intuitive function names for package users as well as simplified maintenance of the code base.

In terms of execution, the **clustering()** function starts by constructing an initial graph layout based on the calculated distances provided through the **scores** argument. In this initial layout, each gene set is a node and edges are drawn between gene sets with a distance smaller than the set **threshold** argument. As edge weights, the distances between the individual gene set pairs are used. The function then applies either the Louvain or Markov clustering algorithm, depending on the value set in the **cluster_method** argument.

For the Louvain clustering, GeDi uses the implementation provided in the R package igraph [Csárdi et al., 2024], which follows the algorithm description in Section 2.3.1. The igraph package is a versatile R package, designed for the creation and analysis of network data. As such, it is widely used across the R and Bioconductor communities, providing efficient and reliable implementations of clustering algorithms such as Louvain.

The output from the igraph implementation is then further postprocessed to create a network where each node represents a gene set, with edges connecting gene sets within the same cluster.

For the Markov clustering, GeDi integrates the implementation of the algorithm provided in the GeneTonic package [Marini et al., 2021]. This implementation was chosen to provide consistency of results across the different R packages developed in our group and to ensure that GeDi seamlessly integrates in the standardised workflow described in our manuscript Ludt et al. [2022]. The algorithm's parameters are set to balance local and global structures in the resulting graph, with both the expansion power p and inflation parameter r fixed at 2. Similar to the Louvain output, the Markov results are used to construct a network of clustered gene sets.

The Fuzzy Clustering is implemented in the **fuzzyClustering()** function, which will take the initial seeds as well as the clustering threshold to determine the resulting clusters. For the identification of the initial seeds, GeDi provides the **seedFinding()** function. Based on the distances of the gene sets as well as set similarity and membership thresholds, the function will determine a list of seeds used for the Fuzzy clustering.

Lastly, the **pamClustering()** function implements the PAM clustering algorithm. The function leverages the implementation of the PAM algorithm in the R package cluster [Maechler et al., 2023]. This will use the calculated distances (provided with the **scores** argument) and the value of k for the number of clusters to determine the final PAM clustering result.

In conclusion, the clustering algorithms available in GeDi provide a powerful toolbox for simplifying and organising functional enrichment results. By grouping gene sets based on similarity, researchers can quickly identify relevant biological patterns, reducing the complexity of large datasets and improving the clarity and coherence of the analysis.

2.4.5 Visualisations of the Clustering Results

In GeDi, clustering algorithms are applied to group gene sets based on their distances. Rather than analysing numerous gene sets simultaneously and needing to keep track of a vast amount of information, clustering reduces this complexity by grouping similar gene sets into a smaller number of clusters. This consolidation allows for easier pattern recognition within the complex data, facilitating the overall interpretation of the results.

To assist the understanding of the clustering results from multiple points of view, GeDi provides various visualisations of the information inherent in the cluster information. The visual representations - a cluster graph, a bipartite graph and a word cloud - offer intuitive ways to explore and interpret the connections between gene sets, assisting the analysis process.

2.4.5.1 Cluster Graph Visualisation

The graph representation uses the unprocessed clustering results, in which each node corresponds to an individual gene set, with edges connecting nodes that belong to the same cluster. This graph provides a visual representation of the connections and rela-

tionships of the gene sets in the input data, as it can easily be observed which gene sets are (functionally) similar.

Figure 10 shows two examples of a graph generated with the **buildClusterGraph()** function of `GeDi`. This function takes clustering results and generates a graph representation of the results. The function also has arguments for the input gene set data, gene set identifiers and gene set description to annotate the resulting graph with node titles and additional information. This is especially useful in interactive version of the graph in the `Shiny` application of `GeDi`, where users can hover over nodes to get additional information (see Figure 14 in Section 2.5.3). Depending on the research question, users can choose which characteristic of the data to visualise in the node colour. This could, for example, be the cluster membership of each node as shown in Figure 10 or other information available in the input data, such as the p-value of the gene set or the number of genes that are associated with the respective gene set the node is representing. It should be noted here that the **buildClusterGraph()** removes singleton clusters from the data before plotting in order to enhance pattern recognition in the data and remove clutter introduced to the graph by a large amount of unconnected nodes. This cluster graph visualisation of `GeDi` is based on and partially makes use of the functions and framework available in the `igraph` R package [Csárdi et al., 2024].

In Figure 10, it can be observed that in the Fuzzy clustering algorithm clusters can be overlapping and share nodes, while the Louvain algorithm exclusively generates cliques (i.e., clusters of nodes in which each node is connected to every other node in the cluster). Although the graphs in Figure 10 are based on the same input distance score, the resulting cluster number and sizes are fundamentally different due to the different characteristics of the used clustering algorithms. The visualisation of these clustering results as a graph facilitates these observation and hence the interpretation of the data.

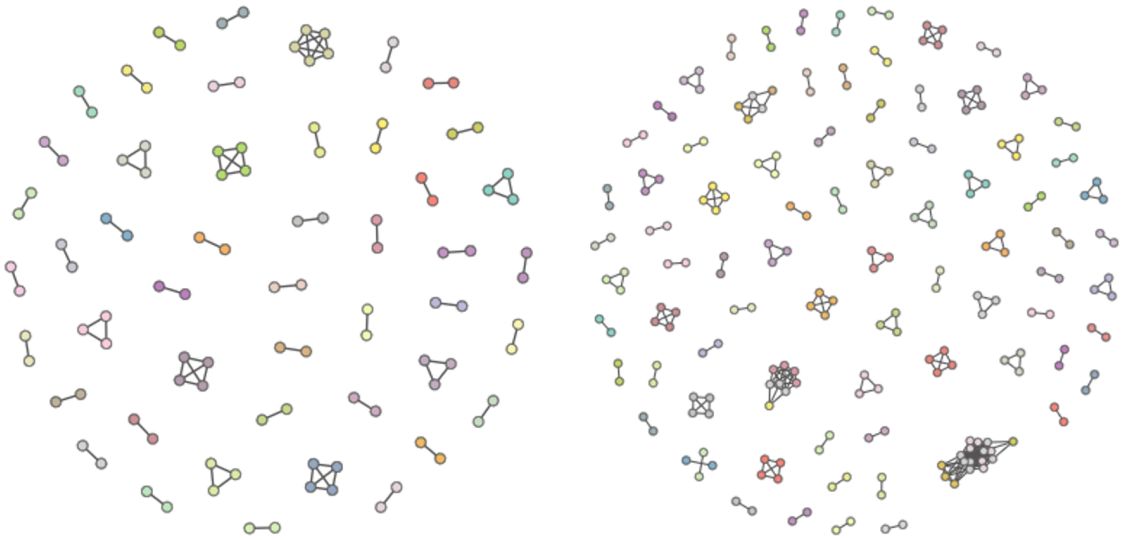


Figure 10 – Graph Representations of Clustering Results
The figure shows two examples of graph representations of clustering results. The gene sets were scored using the Jaccard distance score. On the left side, the clustering result of the Louvain algorithm can be seen, while the right side shows the Fuzzy clustering. In both examples, colours were chosen to indicate the cluster membership.

2.4.5.2 Bipartite Graph Visualisation

The second visualisation available in GeDi for clustering results is a bipartite graph representation. In a bipartite graph, nodes are divided into two distinct groups and edges connect nodes from one group to nodes of the other group. There are no edges between nodes of the same group [Csárdi et al., 2024]. In GeDi, the two node groups are clusters and gene sets and edges are directed from cluster nodes to their respective gene set member nodes.

Figure 11 shows a section of a bipartite graph of clustering results generated with the GeDi function **`getBipartiteGraph()`**. The function uses the clustering results to generate a bipartite graph. The example bipartite graph in Figure 11 shows only a small representative section of a larger bipartite graph to enhance clarity of the example. The example clearly shows how gene sets (the blue, oval shaped nodes in the graph) can be part of several clusters (the yellow, rectangular shaped nodes), indicated by the directional edges of the bipartite graph, which are leading only from cluster nodes to gene set nodes. The bipartite graph can be especially useful to study the relationship between overlapping clusters.

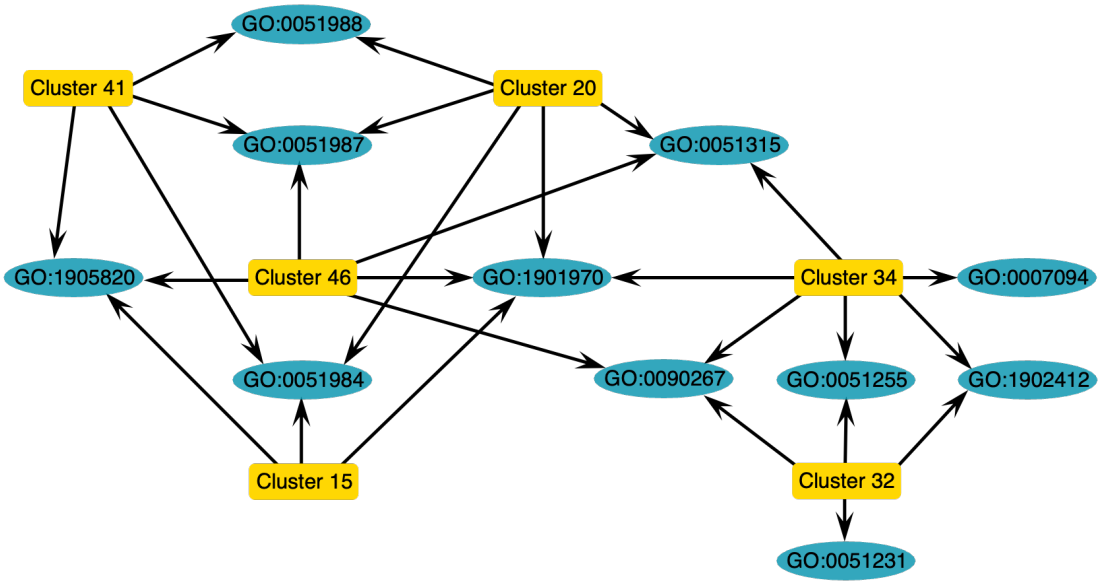


Figure 11 – Bipartite Graph of Clustering Results
The figure shows an example of a bipartite graph representation of clustering results. The nodes in this graph represent both gene sets and clusters. Cluster nodes are coloured in yellow and have a rectangular shape, while gene set nodes are coloured in blue and have an oval shape. Nodes are additionally annotated with either the cluster number for cluster nodes or the gene set ID for gene set nodes. Edges are directed from cluster nodes to their respective gene set member nodes. The figure only shows a rearranged subsection of the original graph to enhance clarity of the figure.

2.4.5.3 Word Cloud Visualisation

The final visualisation option available in GeDi for clustering results is the word cloud. Word clouds are usually used as a visual representation of text data, commonly used to quickly identify prominent themes or terms in datasets. The size of each word in the cloud typically corresponds to its frequency within the text, making word clouds a simple and intuitive way to highlight key concepts in a dataset [Gallery, 2024b].

Figure 12 shows an example of a word cloud for a specific cluster generated with the GeDi function **`enrichmentWordcloud()`**. The input data for this function can either be the complete functional enrichment result data, or, as used in the GeDi Shiny application, the input data subsetted to only the gene sets belonging to a specific cluster. The word cloud is generated based on the description of each gene set.

Functional enrichment analysis results, such as those provided by topGO [Alexa et al., 2006] and `clusterProfiler` [Wu et al., 2021], offer brief descriptions or terms for each gene set, typically corresponding to the pathway names in the database. Examples of these descriptions include general terms like "biological process", "immune response", and "cell death", as well as more specific pathway names like "regulatory T cell differentiation" and "negative regulation of inflammatory response". These descriptions are used to generate the word cloud in GeDi. If these descriptions are not available - they are not a necessary input information for GeDi - the function will use the provided gene set identifiers.

From these terms/identifiers, the function will first generate a corpus of the occurring words and filter common stop words of the English language ("the", "or", "and", etc.) before generating a document-term matrix which includes the frequency of each term. For these steps, GeDi leverages the implementation of the R package `tm` [Feinerer et al., 2008; Feinerer, Hornik, 2024], a package specifically developed for text mining purposes. The resulting document-term matrix is then used to plot a word cloud using the functionality of the `wordcloud` package [Fellows, 2018].

Figure 12 – Word Cloud of an Individual Cluster
The figure shows an example of a word cloud for a specific cluster result. The size of each word reflects its frequency, with larger words such as "mitotic", "regulation" and "spindle" indicating higher frequency and a likely common biological theme in the gene sets of the cluster.

The example word cloud shown in Figure 12 shows the word cloud corresponding to cluster 34 shown in Figure 11. The example indicates that the gene sets belonging to cluster 34 seem to be involved in the process of cell division, potentially related to stages of mitosis. The word cloud vividly summaries the overarching biological themes of the cluster, showing insights into the biological pathways involved.

2.5 The `GeDi` `Shiny` Application

Functional enrichment analysis results are generated at a high and ever increasing speed as they have become an integral part of analyses like bulk-RNA-seq [Subramanian et al., 2005; Wijesooriya et al., 2022]. As discussed in Chapter 1, specifically Section 1.3, these results are usually presented as large, complex lists, with important information scattered over hundreds to thousands of enriched gene sets impeding interpretation and hypothesis generation. Additionally, there is often more than one set of functional enrichment results generated during a single data analysis. This can be due to the complexity of experimental designs or the application of multiple functional enrichment methods to identify the optimal results.

Section 2.4 discussed the individual approaches and functions implemented in `GeDi`, designed to simplify data interpretation and uncover underlying regulatory and interaction patterns. In order to achieve this, `GeDi` is implemented in the R programming language leveraging the rich ecosystem of existing packages and tools available in the Bioconductor community, as well as the CRAN project (`https://cran.r-project.org/`) [Huber et al., 2015; R Core Team, 2024].

In order to further enhance `GeDi`'s functionality as well as increase the accessibility to a broader audience, the package includes a `Shiny` application. The `Shiny` framework is a web application (app) framework, which can be used to create interactive web applications from R code [Chang et al., 2024; R Core Team, 2024]. By utilising `Shiny`, the `GeDi` app offers a dynamic user interface (UI) that allows users to interact with the data and visualisations via various input controls. Thanks to `Shiny`'s reactive programming, the app responds to user input in real time, dynamically updating outputs as changes occur in the underlying data [Chang et al., 2024].

`GeDi`'s `Shiny` application can operate in two modes: It can run locally, which is beneficial for users who prefer offline access (with the prerequisite that PPI data must be downloaded upfront and cached) and who want to leverage local computational resources. Alternatively, it can be deployed on a server, as demonstrated by the demo instance available at `http://shiny.imbei.uni-mainz.de:3838/GeDi`, enabling collaborative use and secure handling of sensitive data. For larger datasets, where computing distance scores can be resource-intensive, switching to server-based execution is recommended, particularly when local hardware capabilities are limited or unavailable.

The following sections will explore `GeDi`'s `Shiny` application. First, Section 2.5.1 will discuss the implementation framework `Shiny`. Next, Section 2.5.2 will provide an overview of the application's user interface. Finally, Section 2.5.3 and Section 2.5.4 will detail the steps taken to ensure that `GeDi` promotes reproducibility of functional enrichment results through interactive data exploration.

2.5.1 The Implementation Framework

`Shiny` is a versatile R package designed to enable the creation of interactive web applications with ease, making data analysis and visualisations more accessible and engaging [Chang et al., 2024; R Core Team, 2024]. While `Shiny` allows users to develop web applications directly in and exclusively with R code, developers can also choose to use HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) or JavaScript, which provides developers with even more flexibility in terms of app design and implementation [Wickham, 2021].

The core principle of `Shiny` is based on reactive programming. Reactive programming describes a programming paradigm which is centered around the automatic propagation of changes, updating output objects whenever their corresponding input objects change. This enables seamless, real-time data and visualisation updates without the need for manual re-execution, allowing users to interactively explore data and instantly see the results of their inputs reflected in visualisations, tables and dashboards [Wickham, 2021].

A typical `Shiny` application is structured around two main components, the user interface and the server, typically implemented in two separated files called `ui.R` and `server.R` [Wickham, 2021; Chang et al., 2024]. The user interface is defined in the `ui.R` file, specifying the layout, input and output elements of the app. Input objects are elements such as sliders, drop-down menus or text fields, which allow the user control over the app's behaviour, while output objects can be visual elements like tables and plots. The content of the `ui.R` file also determines the placement and format of the output objects [Wickham, 2021; Chang et al., 2024].

The `server.R` file, on the other hand, contains the server-side logic, responsible for processing changes to the data objects, handling user input, and generating the corresponding output displayed in the UI [Wickham, 2021; Chang et al., 2024]. This output is automatically updated upon changes to the input, without any need of user intervention. Both components, server and UI, can also be combined in one single `app.R` file, as done in `GeDi`. This approach, however, puts additional requirements on the structure and organisation of the code, because otherwise clarity can be lost among the large amount of code needed for an app [Wickham, 2021].

One of the key advantages of `Shiny` is the tight integration with the R framework [Wickham, 2021; Chang et al., 2024; R Core Team, 2024]. This allows `Shiny` to use the full capabilities of R's data objects and analysis packages within a web-based interface. This enables interactive implementations which use real-time data processing, statistical modelling or machine learning on top of the available data visualisation and exploration options. Thanks to these features, `Shiny` is widely used for creating interactive dashboards, allowing users to analyse, explore and interpret data. This makes `Shiny` particularly useful for various types of exploratory data analysis, where dynamic visualisations of results provide a more intuitive understanding of the dataset.

Furthermore, Shiny applications can be easily deployed online through Shiny Server (https://posit.co/products/open-source/shiny-server/) or shinyapps.io (https://www.shinyapps.io/), enabling the sharing of data exploration tools with broader audiences. This makes Shiny an ideal choice for building data-driven web applications for both private research and public use, promoting open science and collaboration.

Excellent examples of Shiny applications for data exploration, analysis and interpretation include the applications discussed in this thesis - pcaExplorer [Marini, Binder, 2019], ideal [Marini et al., 2020] and GeneTonic [Marini et al., 2021] - as well as the iSEE framework [Rue-Albrecht et al., 2018] and the Magnetique app [Britto-Borges et al., 2022], that I also contributed developing.

2.5.2 Design Principles of the User Interface

The user interface is the point of interaction between a user and a system. In GeDi, the design and structure of the UI are built using the bs4Dash package [Granjon, 2024], which integrates Bootstrap 4 functionality into Shiny applications. This integration enhances both the visual appeal and usability of the dashboard by enabling responsive design and interactive components, such as modals and tooltips. Modals are pop-up elements that appear on top of the main content to display additional information or prompt user actions without navigating away from the current page, while tooltips are small hover-activated text boxes, providing brief explanations or context.

The Bootstrap 4 framework also ensures the interface adapts seamlessly across devices, providing a consistent user experience. The shinydashboard package [Chang, Borges Ribeiro, 2021] complements the UI by offering functionality to build a clean, organised layout. Key UI elements of this layout include the dashboard body for content display and a sidebar menu enabling smooth navigation between panels (Figure 13).

Nearly all features and functionality of GeDi are accessible through the main function, **GeDi()**, which launches the app and directs users to the *Welcome* panel (Figure 13). This panel serves as the main entry point for the application, offering users guidance on how to navigate GeDi, along with detailed explanations and a standardised workflow for preparing data for its use in GeDi. This data preparation mainly refers to the analysis steps necessary to generate functional enrichment results and is tailored specifically to bulk RNA-seq data, following the standardised workflows described by Ludt et al. [2022] and Van Den Berge et al. [2019].

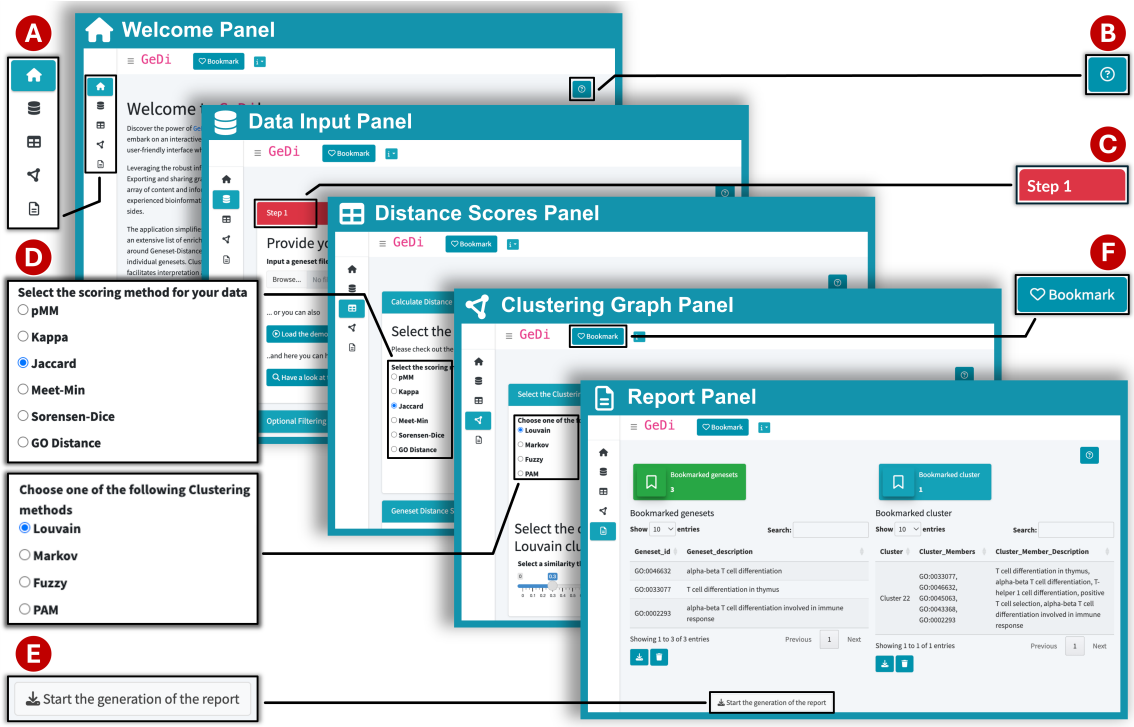


Figure 13 – An Overview of the Available Panels in GeDi
This figure showcases the main panels of the GeDi Shiny application, designed for analysing and visualising functional enrichment analysis results. The figure highlights specific features of the app by providing a zoomed view, such as the sidebar panel (**A**) used to navigate the individual panels of the app or the 'Help' button (**B**), used to activate the interactive tour. Additionally, GeDi features sequentially emerging elements, such as the 'Step 1' box (**C**) and various input selectors throughout the app (**D**). In order to ensure reproducibility, an HTML report can be generated from within the app (**E**), which also highlights interesting findings saved using the 'Bookmark' button (**F**).

From the *Welcome* panel, users can easily navigate to the other panels of GeDi via the sidebar panel (Figure 13A). Designed for intuitive use, the sidebar guides users through the application in a logical, step-by-step sequence that mirrors the workflow described in Section 2.4. The panels are arranged sequentially from top to bottom, beginning with the *Welcome* panel and ending with the *Report* panel, which generally marks the final step of a GeDi analysis session. Aside from the *Welcome* and *Report* panels, users can use the *Data Input* panel for preprocessing steps as outlined in Section 2.4.1, the *Distance Scores* panel for calculating distances and visualisations as described in Section 2.4.2 and Section 2.4.3, and the *Clustering Graph* panel for clustering functionalities detailed in Section 2.4.4 and Section 2.4.5. These panels will be further detailed in Section 3.3, where an example dataset is used to demonstrate a typical use case.

To further enhance usability and provide a dynamic, user-centric learning experience, GeDi includes guided tours that assist users in navigating the app and understanding its features in greater detail. These tours, implemented with the `rintrojs` package [Ganz, 2016], offer an interactive onboarding approach with a step-by-step walkthrough providing detailed descriptions and usage instructions for each panel's component. The tour is accessible in each of GeDi's panels using the question mark (i.e., 'Help') button in the upper right corner of the panel (see Figure 13B). This interactive form of documentation highlights the app's functionality, providing a more engaging learning experience compared to traditional resources, such as package vignettes.

One of the primary targets of this thesis, and a key motivation behind GeDi's development, was to simplify the interpretation of functional enrichment analysis results. These results are commonly used and evaluated by computational and wet-lab biologists to generate hypotheses for future experiments and research questions. Consequently, early interactions, meetings and discussions with collaboration partners were an important pillar in the design process of GeDi to ensure a clean and well-structured UI with a strong focus on user-friendliness. To achieve this, the application employs task-specific panels that guide users through a sequential workflow, making it easy to navigate. Additionally, GeDi features collapsible and progressively revealed elements within the panels, which not only prevent the interface from becoming overwhelming but also enhance the workflow by leading users through data exploration, ensuring they can extract relevant information in a context-sensitive manner (Figure 13C).

Consistency within the app is also achieved through the uniform use of colours and element placement. Clickable buttons are not only distinguished by a consistent colour scheme, but also feature clear, descriptive labels. Reoccurring buttons, such as the 'Help' button (Figure 13B), are strategically placed throughout the individual panels, ensuring a more efficient and intuitive navigation experience.

Shiny's reactive programming allows GeDi to provide immediate feedback to users. During longer computations, progress bars in the lower-right corner (see Figure 6 in Section 2.4.2) keep users informed about the current state of the app. Additionally, loading spinners signal when visualisations are being processed. In rare cases, in which errors are encountered, the underlying implementation is designed not to crash but to alert users with concise error messages. These messages are displayed in the lower-right corner, indicated by red font in order for them to be more easily distinguished from other notifications. The messages include a crisp error description, allowing users to continue their data exploration and avoid future errors of a similar manner.

2.5.3 Interactive Data Exploration

A key component of a variety of research questions and endeavours is the extraction of information from and the subsequent sharing of scientific findings. This usually requires an in-depth analysis of the initial input data or observations, a process that is increasingly supported by advanced computational techniques across a wide range of research fields. Consequently, analysis tools and frameworks have become more accessible to a broader audience, as users can apply the computational methods to their data without the need of understanding and implementing these themselves.

Users are further empowered through the rising number of interactive frameworks implemented for a variety of computational methods. With the increasing amount and complexity of the generated datasets, interactive data analysis processes become more common, helping extract insights from various types of data [Hellerstein et al., 1999; Keim et al., 2006; Battle, Heer, 2019]. Such interactive software and methods should mainly be designed to deliver an intuitive user interaction to increase the accessibility of data analysis to a broader audience.

However, this increasing accessibility could paradoxically become a weakness and potential limitation of these frameworks as the knowledge and understanding of the underlying mathematical and computational methods might become less extensive for the majority of users. In turn, this not only requires a comprehensive and well described documentation of the tools but also strengthened communication between the users and the developing bioinformaticians.

A key element of `GeDi`'s functionality and UI is the interactive environment provided by `Shiny`'s reactive programming in the form of selection inputs, sliders or buttons (Figure 13D). This responsiveness allows users to dynamically adjust parameters and immediately see the effects on visualisations and tables, facilitating a deeper understanding of functional enrichment analysis results.

Additionally, the application incorporates interactive visualisations to further enhance data exploration and interpretation. R packages such as `plotly` [Sievert, 2020] and `visNetwork` [Almende B.V. and Contributors, Thieurmel, 2022] are used to create interactive visualisations, such as the dendrogram and network discussed in Section 2.4.3 and the cluster and bipartite graph discussed in Section 2.4.5. Upon hovering over data points in these visualisations, `GeDi` displays additional details. These details include any additional information that is be available for the gene sets in the input data.

As these details are not a necessary input to `GeDi`, they are not displayed in the visualisations directly, but rather shown upon hovering. This approach maintains a clean and uniform appearance of the visualisations, independent of the availability of additional details, and prevents them from becoming cluttered. Moreover, many visualisations in `GeDi` support zooming and further interactivity, such as reordering the network visualisation by moving and positioning nodes and clusters. An example of the additional information provided by hovering over the visualisation elements can be seen in Figure 14, while Figure 11 from Section 2.4.4 highlights the zooming and reordering feature of the networks.

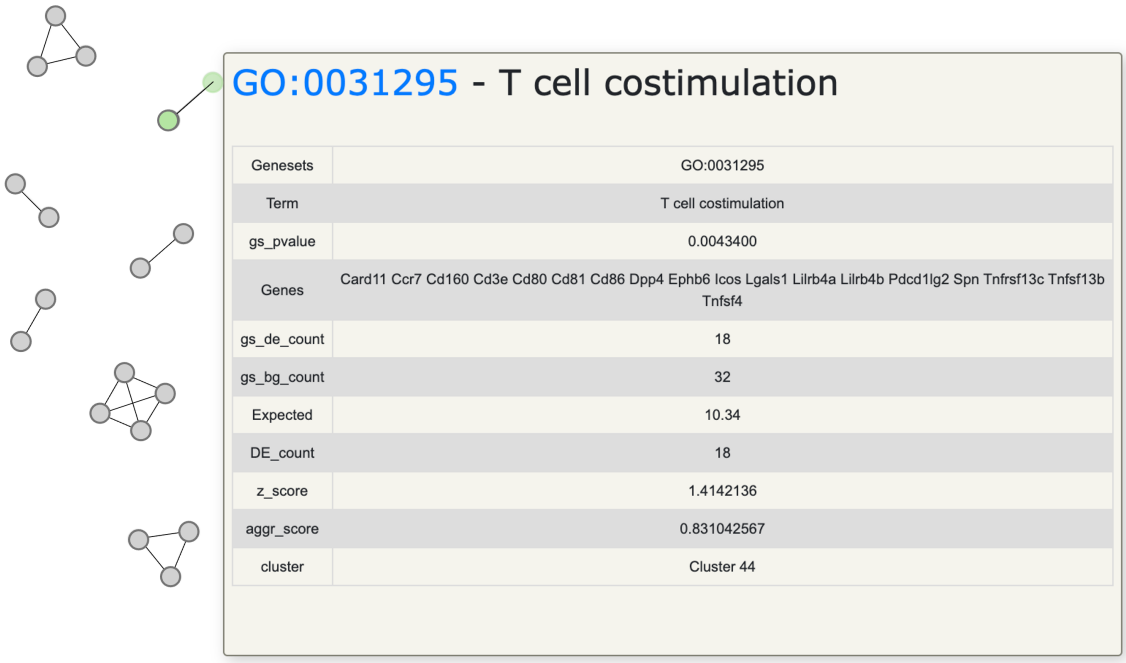


Figure 14 – The Hovering Feature of GeDi
The figure shows an example of the hovering feature in GeDi. The graph displayed is the cluster graph representation of clustering results. Upon hovering over a specific node, additional information available on the input data is displayed. While this feature can be seen in the GeDi application as demonstrated in this figure, the displayed node and edge colours were intensified to enhance the clarity of the figure.

2.5.4 Reproducible Research Practices

In addition to focusing on interactivity and an intuitive user-experience, another key development goal for GeDi was the ability to conveniently reproduce results. With the development of progressively advanced technologies such as bulk and single-cell RNA-sequencing and the corresponding data analysis, the amount of reported scientific findings increases unsurprisingly. However, the pace of these developments has posed challenges, particularly in maintaining comprehensive and transparent documentation of methodologies in published studies. This has also been addressed in various publications, critiquing the current status of reported results across a wide array of studies and publications [Miyakawa, 2020; Munafò et al., 2017, 2022; Wijesooriya et al., 2022].

Although various definitions of the term "reproducibility" or "reproducible research" exist, they share a common foundation: reproducibility refers to the ability of the reader to repeat the presented analysis using the provided data, documentation, and software, in order to replicate the original results [Claerbout, Karrenbach, 1992; Baggerly, Berry, 2011; Stodden et al., 2013a]. Adopting these principles in research would not only lead to a more open science community, but also facilitate collaboration, ensure long-term usability of one's own work as well as support innovation, as clearly documented research is much more likely to be (favourably) reproduced by others [Stodden et al., 2013a,b; Markowetz, 2015; Munafò et al., 2017].

One of the major design principles behind GeDi was to ensure that the exploration of the functional enrichment data is reproducible. Most research endeavours are not the work of a single person, but usually a collaboration between several people, often even

several different research groups of varying expertise. Hence, a straightforward way to share results and findings is often desired by collaborating researchers.

In order to meet this need, GeDi supports the automatic generation of detailed HTML reports of the exploration session (Figure 13E). These reports include dedicated sections corresponding to the individual panels of GeDi, capturing all explorations conducted within the application. The report will start with the information available in the *Data Input* panel, such as a `data.table` of the input data and, if applicable, a `data.table` of the filtered gene sets, information on the species of the input data and the downloaded PPI information. This is followed by a section of the *Distance Scores* panel content, highlighting the selected distance metric as well as providing the three available visualisations of the calculated distances. Afterwards, a section about the *Clustering Graph* panel is included, highlighting the selected clustering algorithm as well as the different visualisations of the clustering results. Thanks to the implementation available in the `visNetwork` package [Almende B.V. and Contributors, Thieurmel, 2022], the network visualisations of the distances as well as the clustering results are also interactive in the HTML report, meaning that hovering over nodes will display the same additional information available in the GeDi Shiny app [Chang et al., 2024] (see Figure 14).

The report feature of GeDi was implemented to not only ensure that users could easily share their findings with collaboration partners and group members, but also reproduce the results of their GeDi session, as the report is a compact summary of the conducted exploration and analysis. As the R and Bioconductor communities are ever improving and developing communities, new versions of R and the packages themselves are released regularly, adding new functionality, but also removing or changing existing functionality. In order to guarantee that the findings achieved using a GeDi session are truly reproducible, the report includes session details at the end, listing all packages used, their installed version numbers, and the local R version of the machine on which the analysis was conducted. With this feature, the report is not only an excellent tool to share findings and results, but also an additional step to ensure reproducibility.

In order to further capture the exploration of the data and support a thorough interpretation of the data, GeDi implements a bookmarking feature. This feature enables users to mark interesting gene sets and clusters for further exploration outside of the application. The bookmarked entities are summarised in the *Report* panel, which provides a list of the bookmarked gene sets and clusters, respectively. To mark individual gene sets and clusters, a bookmarking button is available in the header of the user interface of each panel (Figure 13F). The bookmarked gene sets and clusters are also included in the HTML report that can be generated in GeDi to allow for an in-depth exploration of these interesting findings outside of the Shiny application.

As an additional feature for a further in-depth exploration of the data and results, GeDi supports the download of processed data at various stages within the application. For instance, users can export their input data after removing gene sets, which facilitates future exploration by eliminating the need to repeat the filtering process. Moreover, the downloaded PPI data and the calculated distances can be downloaded and saved to the local machine of the user. This not only provides the possibility to analyse the PPI in-

formation and gene set distances outside of the GeDi framework, but also enables users to share the data and results with collaboration partners and group members.

Furthermore, GeDi implements the principle of reproducible research by not only being available as freely downloadable Bioconductor package, but also by providing the complete implementation code in a public GitHub repository (https://github.com/AnnekathrinSilvia/GeDi). All of GeDi's functionality is available through the GitHub repository, ensuring that the concept behind the individual functions and methods is not applied in a blind, black-box style, but rather fostering an experience in which the user is encouraged to learn about the methods applied to their data. This also promotes and encourages collaboration with fellow R and Bioconductor users and package developers.

Finally, to ensure full reproducibility and transparency of all analyses and results presented in this thesis, several dedicated GitHub repositories were created, each focusing on a specific aspect of this work. A comprehensive overview of all mentioned resources and links to repositories can be found in Appendix A. In particular, an accompanying repository for this thesis was created on GitHub (https://github.com/AnnekathrinSilvia/GSE130842_Showcase), containing a script detailing each analysis step discussed in Chapter 3. This repository also provides the code used to generate several of the figures of this thesis, along with the input data for each presented analysis step and figure, ensuring thorough transparency and reproducibility of the research.

3 Results

The goal of this thesis was to enhance the analysis and interpretation of transcriptome data by developing a standardised workflow using interactive R/Bioconductor packages. For this purpose, I established a standardised analysis workflow using the R packages developed in our research group. This workflow was published in our manuscript Ludt et al. [2022], which provides a step-by-step, well-documented user guide showing how to use the packages to analyse transcriptome data in a robust and reproducible manner.

While developing this workflow, I identified the need for a specialised package focused on the exploration and interpretation of functional enrichment results. As discussed in Section 1.3 and shown in Figure 3, these results are usually presented as large and sometimes redundant lists, in need of an effective approach to be aggregated and visualised. Through an independent literature research, I assessed the need for such a tool by analysing the current reporting and documentation standards of functional enrichment results in published studies.

Consequently, the primary objective of this thesis became the development an intuitive and widely accessible tool to tackle the challenge of functional enrichment interpretation. The result of this effort is the R/Bioconductor package `GeDi`, whose functionality and key design principles I discussed in Chapter 2. In this chapter, I will demonstrate how `GeDi` can facilitate the interpretation of functional enrichment results and identify new and interesting patterns in the data through a showcase of a publicly available dataset.

In Section 3.1, the chapter will begin with a detailed look at standardised workflows for bulk RNA-seq analysis, recapitulating our article "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022]. Subsequently, Section 3.2 will present the results of the literature review, providing insights into current standards and gaps in functional enrichment reporting. Finally, a hands-on demonstration of `GeDi` is presented in Section 3.3, using a publicly available dataset to showcase the tool's capabilities in exploring and interpreting enrichment analysis results interactively.

3.1 Development of Standardised Analysis Workflows

In the scope of this thesis, standardised analysis workflows are defined as a systematic, reproducible approach that ensures consistency throughout the entire data processing pipeline, from raw data processing to the final interpretation of results. Such workflows are essential in research, especially in fields like genomics and transcriptomics, where data complexity requires clear protocols to promote reproducibility, transparency, and efficiency [Girolami et al., 2006; Akhmedov et al., 2020]. By adhering to well-defined steps and using consistent methodologies, researchers can better ensure that their results are reliable and comparable across different studies, allowing for greater scientific precision.

Standardised workflows are particularly valuable in the scope of high-throughput technologies such as RNA-sequencing, where the large volumes and intricate complexity of the data necessitate the adoption of best practices and established standards to ensure accurate and reliable analyses. Implementing a structured and consistent approach helps

optimise data processing, minimise variability, and reduce the risk of errors, leading to more robust and reproducible results. A standardised workflow not only serves as a comprehensive blueprint but also ensures meticulous documentation at each step, which is essential for reproducibility and allowing other researchers to replicate and verify findings. Furthermore, standardised workflows save time by reducing the need for ad-hoc decision making during data processing, while also fostering collaboration by providing a common, predefined workflow that can be used across multiple research teams.

Our article, "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022], provides a comprehensive demonstration of how standardised workflows can be effectively implemented in practice. In this work, we developed an interactive, step-by-step workflow that spans all stages of a typical bulk RNA-seq data analysis, from initial exploration to model building and functional enrichment analysis (Figure 15).

The modelling and downstream analysis of RNA-seq data can be typically divided into four steps, as seen in Figure 2 in Section 1.2.1. These steps are covered by the packages previously developed in our group, `pcaExplorer` [Marini, Binder, 2019], `ideal` [Marini et al., 2020], and `GeneTonic` [Marini et al., 2021], which are designed to seamlessly integrate with one another, as well as with the broader R/Bioconductor ecosystem (Figure 15). Each package focuses on distinct, yet complementary steps of the analysis, ensuring that researchers can easily adopt the packages either individually or as a complete workflow into their own projects, as seen in the works published by others in the scientific community [Ahmad et al., 2022; Monga et al., 2022; Kim et al., 2023; Pilz et al., 2023; Olechnowicz et al., 2024].

The workflow presented in our article aimed to make bulk RNA-seq analysis accessible to a broader audience. To achieve this, we divided the article into three *Basic Protocols*, each centered around different steps of the modelling and downstream analysis (Figure 15). Each *Basic Protocol* focuses on one of the three packages, providing detailed, step-by-step instructions to guide users through the analysis process, demonstrating how to effectively use each package. For this purpose, the protocols leverage the `Shiny` applications available in `pcaExplorer`, `ideal` and `GeneTonic`. These interactive applications allow users to dynamically explore and analyse their data, making the process accessible even to those with limited coding experience.

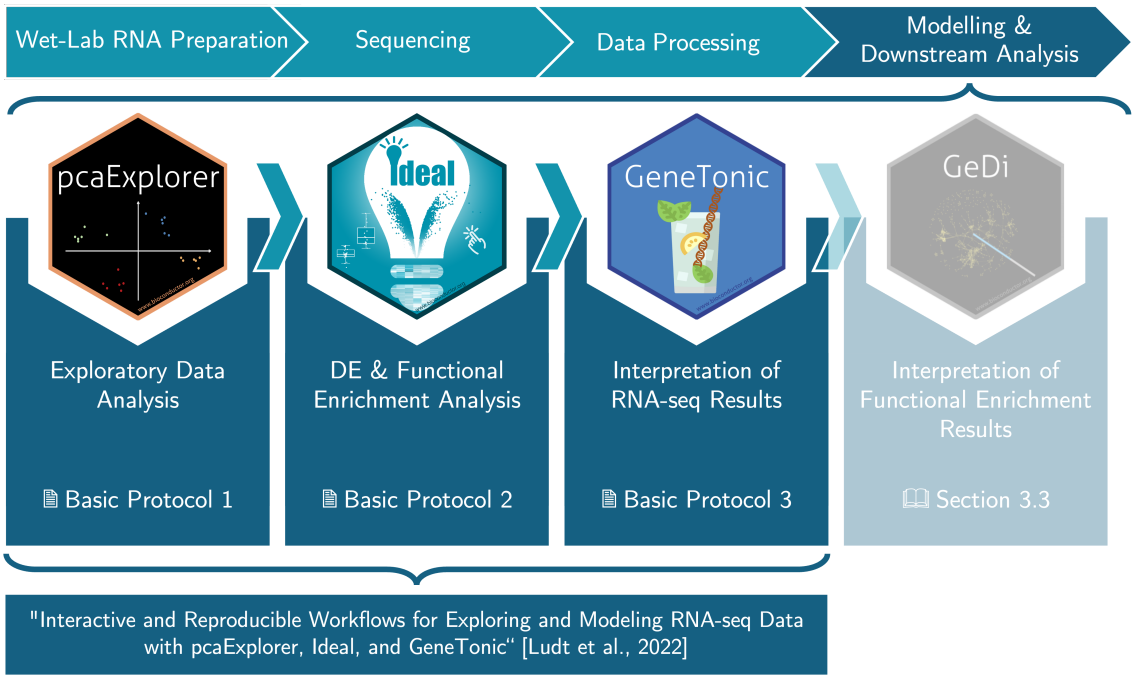


Figure 15 – Overview of the Standardised Analysis Workflow for Transcriptome Data
The figure shows the typical bulk RNA-seq analysis workflow from wet-lab preparation of the RNA to the modelling and downstream analysis. It highlights how the individual steps of the modelling and downstream analysis are covered by the R packages developed in our group and the standardised workflow developed and published in Ludt et al. [2022]. This workflow will be extended in this thesis through the newly developed package `GeDi`.

In *Basic Protocol 1*, we demonstrated how to use the `pcaExplorer` package [Marini, Binder, 2019; Ludt et al., 2022] for exploratory data analysis. `pcaExplorer` is an interactive R package that enables users to explore the Principal Component (PC) latent space of the data. Through its interactive Principal Component Analysis (PCA), `pcaExplorer` helps identify outliers in the samples as well as inspect the relationships between samples. In this protocol, we demonstrated how to generate various visualisations from the count data to identify patterns, outliers and sample relationships of the data using the `pcaExplorer Shiny` application. This EDA process aids in quality assessment and hypothesis generation, facilitating a deeper understanding of the RNA-seq data at hand. The protocol also details how to prepare the necessary input for `pcaExplorer` from the count matrix to produce a `DESeqDataset` [Love et al., 2014] object — a standardised data format widely used in the R and Bioconductor communities [Ludt et al., 2022]

Basic Protocol 2 focuses on conducting differential expression analysis using `ideal` [Marini et al., 2020], which is built on the `DESeq2` [Love et al., 2014] framework. In this protocol, we guided users through data preparation, model specification, extraction, and interpretation of DE results [Ludt et al., 2022]. It covers setting up the necessary objects from the count data, specifying experimental designs, running the DE analysis and exploring the results through visualisations like MA plots and volcano plots. These plots visualise the relationship between the fold change (FC), i.e., how strongly gene expression changes between samples, and either the mean expression or the statistical significance of the genes. We also included a demonstration of functional enrichment analysis to identify over-represented biological pathways among the differentially expressed genes; a commonly performed step in downstream analyses of bulk RNA-seq data as well as other omics datasets [Reimand et al., 2019; Geistlinger et al., 2021].

Finally, *Basic Protocol 3* provides a guide for integrating and interpreting RNA-seq analysis results using the `GeneTonic` package [Marini et al., 2021; Ludt et al., 2022]. `GeneTonic` enables users to combine DE analysis results with functional enrichment data, offering a comprehensive overview of the biological processes and pathways affected in the study. In this protocol, we included steps for data integration, gene set visualisation, and contextualisation of RNA-seq results in biological terms, aiding in a more thorough understanding of the underlying biology.

Additionally, the article includes a *Support Protocol* and an *Alternate Protocol*. The *Support Protocol* describes in detail how to install the necessary software packages and download the example data presented in the article. The *Alternate Protocol* provides instructions how to integrate the functionality of the packages into custom scripts and analyses [Ludt et al., 2022].

In summary, our article demonstrated how the presented standardised workflow enhanced reproducibility, interactivity, and efficiency of a bulk RNA-seq data analysis. Using our packages `pcaExplorer`, `ideal` and `GeneTonic`, the workflow ensured that each step of a typical downstream data analysis was fully documented and reproducible, leading to consistent and reliable results. Moreover, the included `Shiny` applications promoted a more comprehensive and interactive exploration of the data, making the workflows accessible to users with various expertise levels.

By combining our packages into the presented standardised workflow, the article supports researchers in implementing best practices in bulk RNA-seq analysis, ultimately enhancing the quality and reliability of their analyses and decision-making.

3.2 Reporting Standards in Functional Enrichment Analysis: A Literature Review

During the work on our article "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022], we identified a substantial gap in the availability of interactive and comprehensive tools for the interpretation of functional enrichment results. In order to determine whether this need was unique to our group or indicative of a broader issue within the scientific community, I conducted a literature review.

This review aimed to assess the current state of functional enrichment reporting in published research and evaluate the significance and urgency of addressing this gap with proper tools and methods. Overall, a total of 97 articles were selected for review using various search criteria and a random sampling method, as outlined in Section 2.1. A list of the selected articles can be found in Appendix B. Additionally, the code for the selection and random sampling can be found on GitHub (`https://github.com/imbeimainz/HIPSTER`) and a detailed table of the results discussed below can be found on Zenodo (`https://doi.org/10.5281/zenodo.13843024`).

The first literature review involved 20 randomly selected articles, of which 16 reported functional enrichment analyses following the description in Section 1.3 (see Figure 16A)[4]. Of the remaining four, two did not include any functional enrichment results, one article had unavailable supplementary material, leading to missing method documentation, and one article was written in Chinese, preventing accurate evaluation due to the language barrier.

Among the 16 articles that included functional enrichment results, the level of detail in documentation varied significantly: two articles (12.5%) provided no details at all, two (12.5 %) documented their methods comprehensively and the remaining 12 (75%) gave only broad descriptions insufficient for reproducing the analysis (Figure 16A). A similar trend was observed regarding software documentation, where 13 articles (81.25%) reported the software used, but only one specified the exact function applied (Figure 16B & C).

The majority of articles (87%) used GO and/or KEGG terms for their functional enrichment analysis (Figure 16D), and all but one visually presented the results, typically in the form of enrichment plots (Figure 16E). Half of the articles also provided tables containing the enrichment results (Figure 16F). Additionally, every article reviewed highlighted a specific subset of gene sets from their functional enrichment analysis (Figure 16G), though three failed to justify their selection criteria. Selection was often justified by presenting the "top" gene sets, which were likely those with the lowest p-values (Figure 16H). Finally, only two articles mentioned aggregating enrichment results through clustering (Figure 16I).

[4]For all figures shown in Section 3.2, the sum of articles displayed in panels **B**, **C**, **D**, **E** and **H** can exceed the overall number of articles reviewed in the specific round of literature review. This is due to the fact that several articles presented multiple, distinct entries for the displayed parameter. Each occurrence was counted individually, leading to totals greater than the number of articles screened.

Figure 16 – Results Overview of the Manual Pubmed Search
The bar plots of the figure illustrate the criteria used to evaluate the reproducibility of the results, such as the reported degree of detail in the documentation of the enrichment methods (**A**), the software tools used for the analysis (**B**), the statistical tests or functions applied (**C**) and the gene set identifiers used in the results (**D**). Additionally, it shows the types of visualisations presented (**E**), whether the results were provided in a table (**F**), whether a specific subset of gene sets was highlighted (e.g., discussed in the text or presented visually) (**G**), the justification for gene set selection (**H**), and whether the enrichment results were aggregated or clustered (**I**).

The second round of literature review was conducted using the R package pubmedR [Aria, 2020]. As documented in the corresponding GitHub repository (`https://github.com/imbeimainz/HIPSTER`), the package was used to search for articles containing the keywords "Enrichment" and "Analysis", published in any journal in 2023. Additionally, as a consequence of the first review, the search was further restricted to articles written in English. This yielded 21,270 articles, from which the first 9,999 were downloaded via pubmedR. Afterwards, 20 were randomly selected. Since 9 of these 20 articles did not feature functional enrichment methods, additional articles were sampled until 20 eligible ones were reached, leading to a total of 33 reviewed articles (Figure 17A).

Once again, the documentation quality varied significantly, with three articles (15%) providing no method details at all, two (10%) featuring comprehensive documentation, and the remaining 15 (75%) only providing general descriptions insufficient for replicating the analysis (Figure 17A). On the other hand, the reporting of the software used was more thorough, with 17 (85%) articles reporting on the used software (Figure 17B).

However, this was not reflected in the documentation of the statistical test or function used, which were only mentioned in two instances. Out of these, one specified the exact function applied, while the other stated a manual curation of the results (Figure 17C).

The most commonly used gene set identifiers were again GO and KEGG terms, with numerous articles using both (Figure 17D). Enrichment plots were the primary method of visualising results (Figure 17E), and more than half of the evaluated articles (60%) also provided their results in the form of a table (Figure 17F). Additionally, every article reviewed highlighted a specific subset of gene sets from their functional enrichment results (Figure 17G), though more than half (65%) of these failed to justify their selection criteria. The selection was often justified by presenting the "top" gene sets or the ones with the smallest p-value (Figure 17H). In contrast, one article mentioned to specifically focus on new results, i.e., gene sets which have not yet been covered in relevant literature of their research field. Finally, six articles mentioned aggregation of enrichment results through methods like clustering (Figure 17I).

Figure 17 – Results Overview of the Search using pubmedR
The bar plots of the figure illustrate the criteria used to evaluate the reproducibility of the results, such as the reported degree of detail in the documentation of the enrichment methods (**A**), the software tools used for the analysis (**B**), the statistical tests or functions applied (**C**) and the gene set identifiers used in the results (**D**). Additionally, it shows the types of visualisations presented (**E**), whether the results were provided in a table (**F**), whether a specific subset of gene sets was highlighted (e.g., discussed in the text or presented visually) (**G**), the justification for gene set selection (**H**), and whether the enrichment results were aggregated or clustered (**I**).

Lastly, the pubmedR package was used to perform a final round of literature review, specifically targeting articles published in the journals *Cell*, *Nature* and *Science*. By focusing on these prestigious, high-impact, and rigorously peer-reviewed journals, the review aimed to assess methodological standards across various research fields, as publications in these journals often influence and shape best practices within the broader scientific community. The focused search resulted in 44 articles, which were all reviewed. 24 of these could not be evaluated as they did not include functional enrichment results as described in Section 1.3. The remaining 20 included either broad descriptions (55%) insufficient for replicating the analysis or very detailed descriptions which ensured reproducibility (Figure 18A). Additionally, all but three (85%) mentioned the software used for the analysis, with half also specifying the statistical test or function used to calculate the results (Figure 18B & C).

The most common identifier used in the functional enrichment analyses were GO terms (Figure 18D). In this review round, all articles visually presented the results, typically in the form of an enrichment bar plot (Figure 18E). More than half of the articles (55%) also provided tables containing the enrichment results (Figure 18F). Additionally, every article reviewed highlighted a specific subset of gene sets from their functional enrichment results (Figure 18G), though half failed to justify their selection criteria. The remaining articles usually supported their selection through field relevant literature or based it on p-values (Figure 18H). Finally, five articles mentioned aggregating enrichment results through clustering (Figure 18I).

Figure 18 – Results Overview of the Focused Search in High Impact Journals
The bar plots of the figure illustrate the criteria used to evaluate the reproducibility of the results, such as the reported degree of detail in the documentation of the enrichment methods (**A**), the software tools used for the analysis (**B**), the statistical tests or functions applied (**C**) and the gene set identifiers used in the results (**D**). Additionally, it shows the types of visualisations presented (**E**), whether the results were provided in a table (**F**), whether a specific subset of gene sets was highlighted (e.g., discussed in the text or presented visually) (**G**), the justification for gene set selection (**H**), and whether the enrichment results were aggregated or clustered (**I**).

The evaluation of the individual rounds of literature review already showed a gap and shortcoming in the reporting of functional enrichment analysis results. In order to further evaluate this, the individual completeness and quality of the selected articles was evaluated.

For this purpose, I first excluded all non-eligible articles, i.e., the ones which did not report functional enrichment results. Afterwards, each article was assessed across the parameters evaluated above and a scoring system was employed, where each article received a "completeness score" based on whether it provided sufficient information for each criterion. Certain values, such as "Not applicable" or "Not stated", indicated missing or insufficient information and resulted in no points being awarded for that criterion. In contrast, valid entries contributed to the overall score. For the specific criterion of whether the article included detailed information on enrichment analysis, points were only awarded if the method was thoroughly described.

This completeness score provided a quantitative measure of how well each article adhered to best practices in documentation and reporting. The resulting score ranged from 0 to 9, with higher scores indicating higher levels of completeness in the information provided.

In Figure 19, a histogram of the distribution of these completeness scores can be seen. The histogram shows that the majority of articles received scores between 4 and 7, indicating varying levels of thoroughness in their documentation. Notably, two articles had a score of 3, while only two articles fully met all criteria, achieving a score of 9.

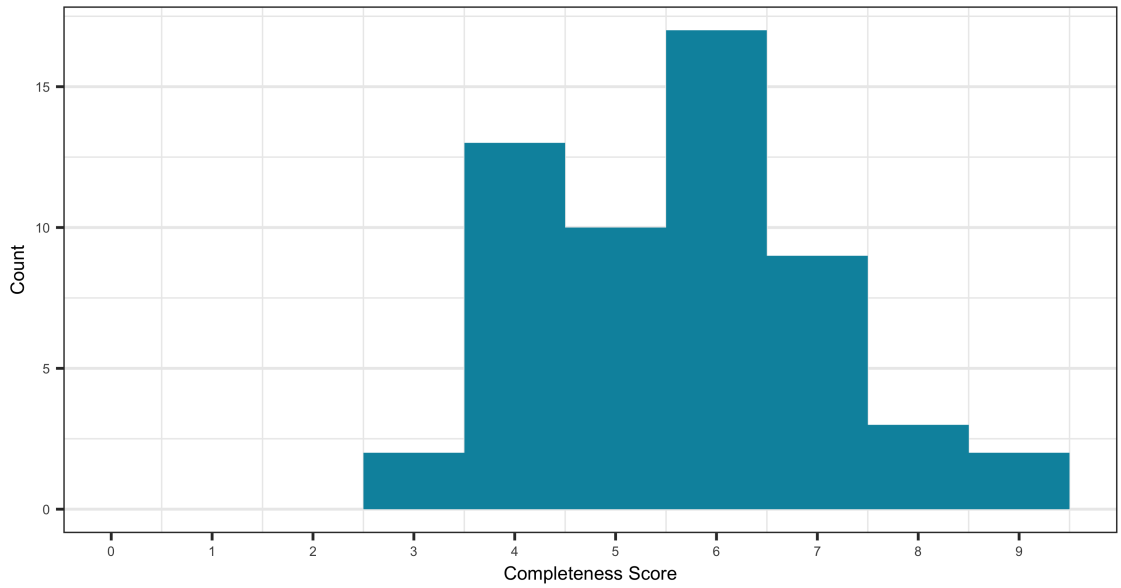


Figure 19 – Distribution of Completeness Scores
The figure illustrates the distribution of scores assigned to the evaluated articles, reflecting their level of completeness in reporting functional enrichment analysis methods and results. The abscissa represents the range of the completeness score, while the ordinate indicates the number of articles that achieved each score.

Overall, this review highlighted significant shortcomings in the documentation of functional enrichment analyses across a wide range of studies. Many articles failed to provide sufficient information for reproducibility, often omitting crucial details about the software and statistical tests used. This trend was somewhat less pronounced in articles published in *Cell*, *Nature*, and *Science*, which exhibited a better standard of reporting.

However, not all of the articles published in these journals ensured the reproducibility of their results, with 55% failing to provide the necessary information on the statistical test or functions used in the analysis, as well as the rationale behind the selection of highlighted results, which were missing in 50% of the articles. Nearly all of the reviewed and eligible articles (93%) presented their results in some form of visualisation, typically in the form of enrichment plots or maps. These visualisations were often focused on a selected subset of gene sets, though justifications for these selections were missing in nearly half (46%) of the evaluated articles. Only around 23% of articles aggregated results to simplify interpretation.

These findings highlight the urgent need for more robust tools to streamline the exploration and interpretation of functional enrichment results. The conclusions of the

article of Wijesooriya et al. [2022] remain at the time of writing as relevant as at the time of publication, as the lack of proper reporting and documentation still hampers reproducibility of the presented analyses. GeDi aims to address this gap, offering a more intuitive and interactive way to interpret functional enrichment results, which could ultimately improve reporting standards and the overall quality of research in the field.

3.3 Showcasing GeDi's Functionality on Publicly Available Transcriptome Data

In order to demonstrate how GeDi facilitates the exploration and interpretation of functional enrichment results, the following sections showcase GeDi's functionality on publicly available murine bulk RNA-sequencing data. Additionally, the section will also highlight, how GeDi seamlessly integrates in and extends our standardised RNA-seq analysis workflow to support an in-depth analysis of functional enrichment analysis results (Figure 20).

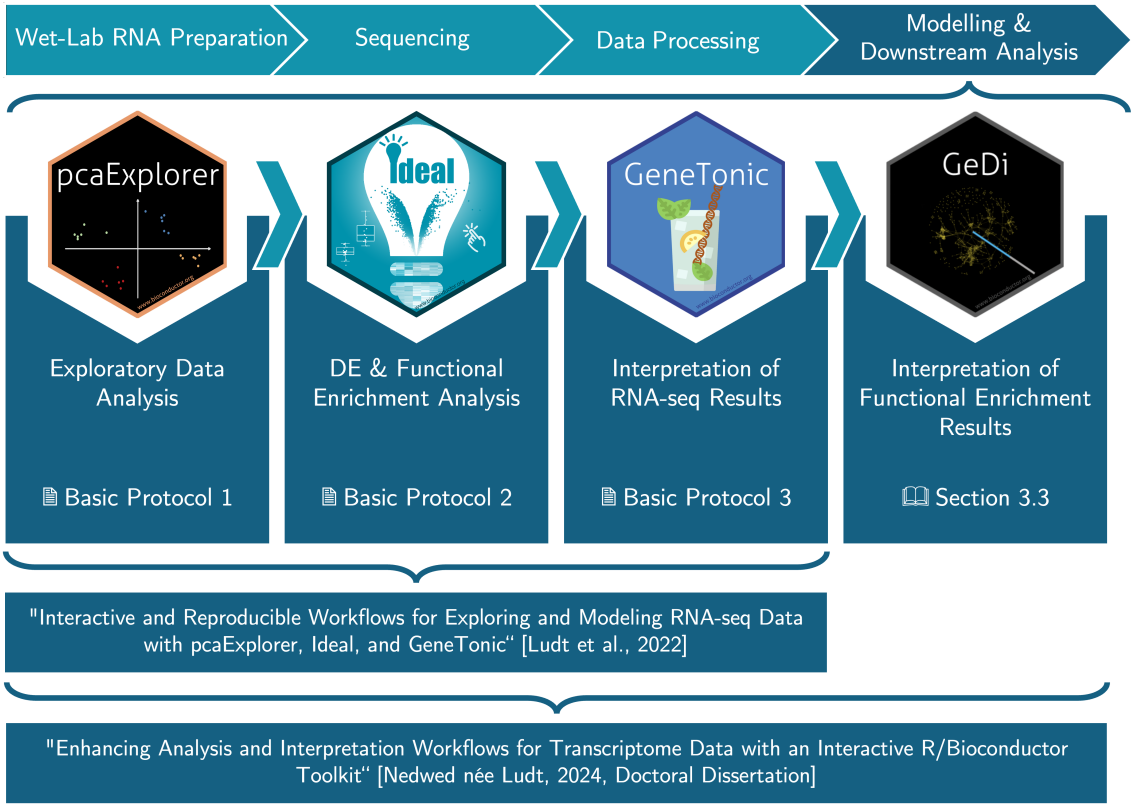


Figure 20 – Extending the Standardised Analysis Workflow with GeDi
The figure shows the typical bulk RNA-seq analysis workflow from wet-lab preparation of the RNA to the modelling and downstream analysis. It highlights how the individual steps of the modelling and downstream analysis are covered by the R packages developed in our group and the standardised workflow developed and published in Ludt et al. [2022]. The figures also portrays how the work presented in this thesis extends our published article with GeDi to enhance the interpretation of functional enrichment results.

First, Section 3.3.1 will first introduce the dataset used for this purpose. This dataset consists of murine bulk RNA-seq data, which was first published in Delacher et al. [2020]. Using the standardised workflow of our manuscript [Ludt et al., 2022], the data is then analysed and prepared for its use in GeDi in Section 3.3.2 and Section 3.3.3. Finally, Section 3.3.4 illustrates GeDi in action, covering the data preparation, distance scores

calculation (Section 3.3.4.1), clustering (Section 3.3.4.2), and the report generation feature (Section 3.3.4.3). Throughout Section 3.3.4, I will also highlight and explore the different available distance metrics and clustering algorithms, showcasing their individual properties and subsequent results.

All of the analyses and processing described in the following sections can also be found on GitHub (`https://github.com/AnnekathrinSilvia/GSE130842_Showcase`) to ensure reproducibility of the described results.

3.3.1 Utilised Dataset: Transcriptome Data of Murine Regulatory T Cells

Regulatory T (Treg) cells are a specialised subset of T cells primarily responsible for immune-regulatory functions [Sakaguchi et al., 2008]. These cells are essential in suppressing self-reactivity and preventing excessive inflammation by modulating a variety of immune cells [Sakaguchi et al., 2008; Delacher et al., 2020]. In non-lymphoid tissues, Tregs can accumulate and perform homeostatic and regenerative functions [Delacher et al., 2020].

However, it has long been unclear whether common precursors for non-lymphoid Treg cells exist and how these cells differentiate. In their study, Delacher et al. investigated non-lymphoid Tregs and identified Batf (Basic leucine zipper transcription factor, ATF-like) as the driver of the molecular tissue program in their precursor cells [Delacher et al., 2020]. Among other data types analysed in the study, the authors generated bulk as well as single-cell RNA-sequencing data of Tregs to compare the gene expression profiles of different Treg subpopulations [Delacher et al., 2020]. These analyses provided valuable insights into the differentiation pathway of Treg populations.

The bulk RNA-seq data presented in the manuscript consisted of 56 samples of T cells isolated from various murine tissues. Cells were isolated ex-vivo from Nfil3 (Nuclear factor interleukin 3 regulated) GFP (Green Fluoresecent Protein) reporter animals or C57/BL6 wildtype animals. The cells included in-vitro expanded and differentiated T cells as well. The analysed tissues included spleen, bone marrow (BM), fat, skin, lung and liver. From these tissues, the authors isolated Killer cell lectin-like receptor subfamily G member 1 (Klrg1)-negative Suppression of Tumorigenicity 2 (ST2)-negative or Klrg1-positive ST2-positive Tregs[5]. In addition, the authors isolated Klrg1-positive Nfil3(GFP)-positive Tregs, Klrg1-negative Nfil3(GFP)-positive Tregs and Klrg1-negative Nfil3(GFP)-negative Tregs from spleens of Nfil3(GFP) reporter animals. The data is available on the Gene Expression Omnibus (GEO) [Edgar et al., 2002; Barrett et al., 2013] under accession number GSE130842 (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE130842`) [Delacher et al., 2020].

Following discussions with the original authors, I decided to focus on the Klrg1-negative Nfil3(GFP)-negative Treg and Klrg1-positive Nfil3(GFP)-positive Treg samples to demonstrate the advantages of standardised workflows and showcase the functionality and benefits of `GeDi` in this thesis. The samples will be referred to as $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and

[5] These samples are called *tisTregST2_Spleen*, *tisTregST2_BM*, *tisTregST2_Fat*, *tisTregST2_Skin*, *tisTregST2_Liver* and *tisTregST2_Lung* in the original Gene Expression Omnibus entry of the dataset under accession number GSE130842 (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE130842`).

$\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ Treg samples from now on.

In order to prepare the data for the following showcase, the available count table was downloaded from GEO (latest access: March 8th, 2022) and processed with R [Edgar et al., 2002; Barrett et al., 2013; R Core Team, 2024]. The data was imported into an active R session using the `read_excel()` function from the `readxl` package [Wickham, Bryan, 2023]. The data was then filtered to include only the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ Treg samples, resulting in a dataset of eight samples used for the subsequent analysis.

An R `data.frame` object containing metadata information was created, including the condition of each sample (either $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ or $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$) and the replicate number as indicated in the GEO entry of the original data (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE130842`) [Delacher et al., 2020]. This metadata, along with the filtered count data, was used to set up a `DESeqDataSet` object [Love et al., 2014], which serves as the initial input for the data analysis described in the following sections.

The design, i.e., the characteristic across which the gene expression is compared, was set to the samples' condition. Additionally, the data was normalised in this preprocessing to adjust for the sequencing depth of the individual samples. Without normalization, differences in gene expression levels might reflect unequal sequencing depths rather than true biological variation. For this, the `estimateSizeFactors()` function from the `DESeq2` package was used [Love et al., 2014]. After this preprocessing, the data was ready for the modelling and downstream analysis steps of the bulk RNA-seq analysis workflow.

3.3.2 Exploratory Data Analysis using `pcaExplorer`

Exploratory Data Analysis is an important step in various omics data analyses. Before the data can be analysed in detail, it is essential to obtain a general overview of the quality of the data and the correlations between the samples. In our original manuscript, "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022], we proposed a standardised workflow for bulk RNA-sequencing data analysis including the use of `pcaExplorer` for the exploratory data analysis in *Basic Protocol 1*. This section will follow the workflow proposed in *Basic Protocol 1* to perform EDA on the Treg bulk RNA-seq data and asses the quality of this dataset.

Section 3.3.1 discussed how the count data used in this thesis was obtained and preprocessed. Before loading this data into `pcaExplorer`, an annotation data object was created. This `data.frame` object was used to map gene identifiers (IDs) from the ENSEMBL database [Yates et al., 2020] to the more commonly used HGNC gene symbols [Seal et al., 2023]. While this step is optional in the suite of `pcaExplorer` [Marini, Binder, 2019; Ludt et al., 2022], it can enhance the overall user experience and facilitate data interpretation. Researchers are generally more familiar with the HGNC gene symbols of their gene(s) of interest, making data exploration and interpretation more intuitive. The annotation object was generated using the `get_annotation_orgdb()` function from

the pcaExplorer package [Marini, Binder, 2019], following the standardised workflow proposed in our article [Ludt et al., 2022].

Once both the data and annotation object were prepared, they were provided to the pcaExplorer() function. This function launched an instance of the pcaExplorer app, displaying the *Data Upload* panel (Figure 21). Typically, in this panel, user can provide the count data as well as the metadata to the application, if not already done so upon the call of the pcaExplorer() function.

Additionally, this panel enables users to generate the required DESeqDataSet object via buttons, eliminating the need for a scripted generation of the object as discussed in Section 3.3.1. In the case of this thesis, the scripted generation of the object was chosen because the obtained count data had to be subsetted for the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples. Hence, the scripted generation of the object was a slightly more straightforward solution than the creation via the pcaExplorer app. Readers find both entry points to pcaExplorer thoroughly documented in Ludt et al. [2022].

Using the button 'Compute variance stabilized transformed data from the dds object' (Figure 21A), the variance stabilized transformation of the input count data was calculated. This transformation is a statistical method used to stabilize the variance of the data across all levels of expression. This is used to avoid biases which could arise from the heteroskedasticity of the data.

In bulk RNA-seq data, genes with overall higher expression levels tend to show more variation in their expression compared to genes with lower mean expression values, leading to heteroskedasticity. Ultimately, this can lead to biased or misleading results in downstream analyses such as PCA, as heteroskedasticity can cause the analysis to be dominated by a few highly variable genes, which might not be representative for the overall difference across samples and conditions, thus overshadowing true biological signals. Variance stabilizing methods, such as those implemented in pcaExplorer, lead to more homoscedastic data. This ensures that downstream analyses become more robust, therefore, reflecting meaningful, biological patterns [Love et al., 2015].

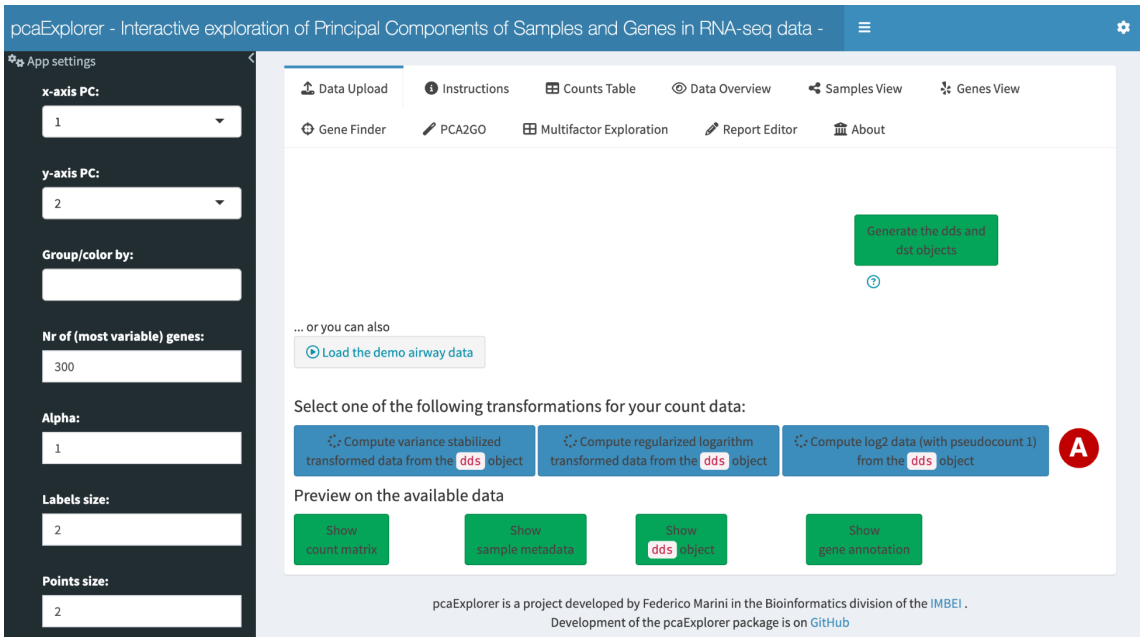
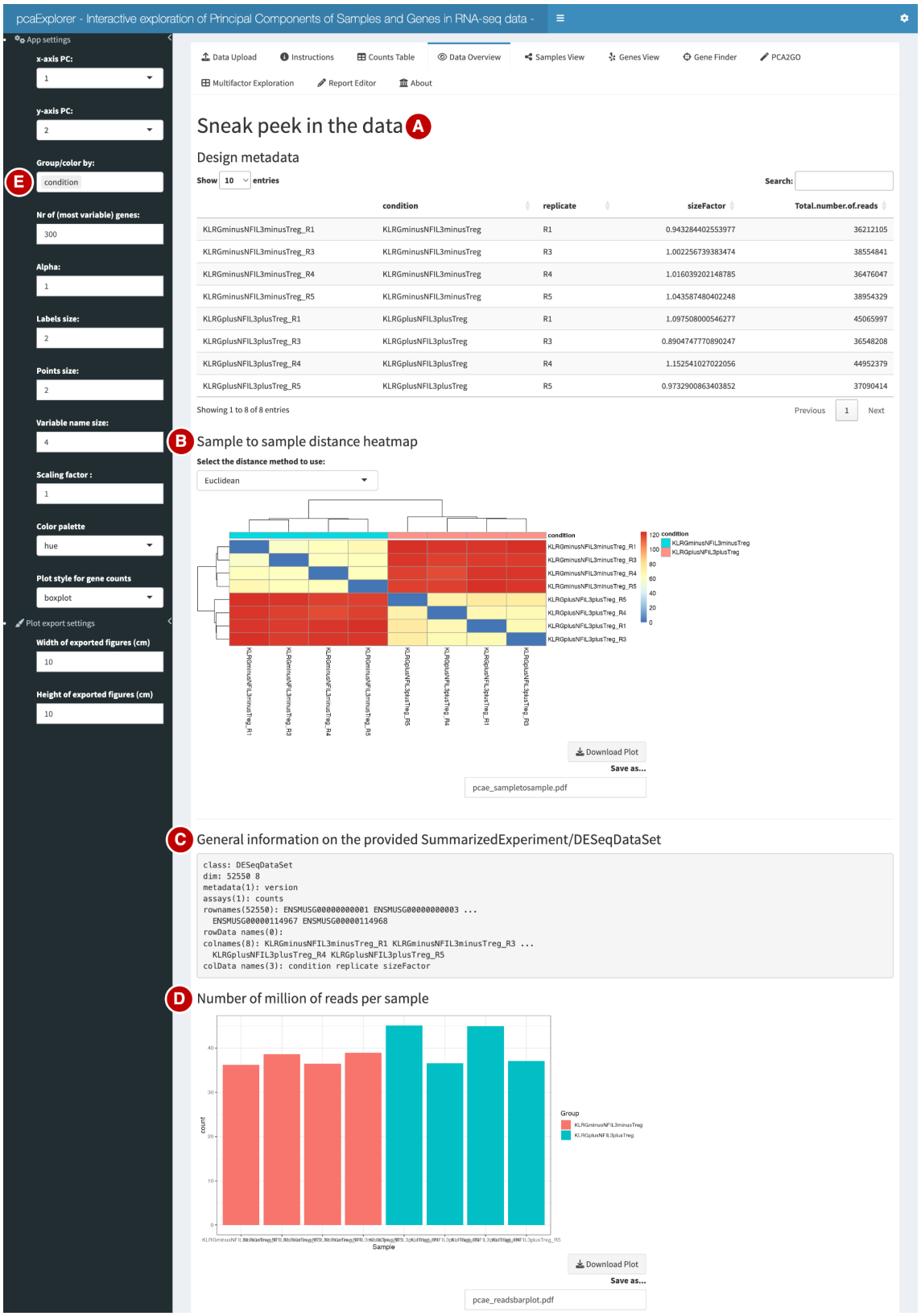


Figure 21 – The *Data Upload* Panel of `pcaExplorer`
The figure shows the *Data Upload* panel of the `pcaExplorer` application, which can be used to provide the data to the application and calculate variance stabilizing transformations of the data (**A**).

Based on the workflow outlined in *Basic Protocol 1* of our manuscript [Ludt et al., 2022], the next step involved exploring the *Data Overview* panel. The panel provided a table of the available metadata information as well as a sample-to-sample heatmap (Figure 22A & B). The sample-to-sample heatmap used a colour gradient to visualise the correlation or similarity between the different dataset samples based on the their gene expression levels. For the Treg dataset, the heatmap indicates a higher similarity among samples within the same condition compared to the inter-condition similarity (Figure 22B). Given the design of the `DESeqDataSet` object and the data preparation described by Delacher et al. [2020], I anticipated that the sample condition would be the primary driver of dissimilarity within the dataset.

Further down in the panel, `pcaExplorer` provided general information about the dataset (Figure 22C), such as the total number of genes (52,250) and samples (8) in the dataset. Below this general information, a bar plot displayed the number of sequencing reads per sample, expressed in millions (Figure 22D). The bar plot shows that the Treg dataset has consistent reads counts across all samples, ranging from approximately 35 to 45 million reads per sample. This indicates that the samples are of good quality, as uniform sampling depths are important for reliable and unbiased comparisons between samples conditions.

Additionally, to enhance the visualisations across this and all following panels, I used the sidebar panel to colour and group the visualisations by the condition of the samples (Figure 22E). This enhanced the interpretation of the visualisations by highlighting the individual sample groups in the Treg dataset with distinct colours, making them easier to distinguish.



	condition	replicate	sizeFactor	Total.number.of.reads
KLRGminusNFIL3minusTreg_R1	KLRGminusNFIL3minusTreg	R1	0.943284402553977	36212105
KLRGminusNFIL3minusTreg_R3	KLRGminusNFIL3minusTreg	R3	1.002256739383474	38554841
KLRGminusNFIL3minusTreg_R4	KLRGminusNFIL3minusTreg	R4	1.016039202148785	36476047
KLRGminusNFIL3minusTreg_R5	KLRGminusNFIL3minusTreg	R5	1.043587480402248	38954329
KLRGplusNFIL3plusTreg_R1	KLRGplusNFIL3plusTreg	R1	1.097508000546277	45065997
KLRGplusNFIL3plusTreg_R3	KLRGplusNFIL3plusTreg	R3	0.8904747770890247	36548208
KLRGplusNFIL3plusTreg_R4	KLRGplusNFIL3plusTreg	R4	1.152541027022056	44952379
KLRGplusNFIL3plusTreg_R5	KLRGplusNFIL3plusTreg	R5	0.9732900863403852	37090414

Figure 22 – The *Data Overview* Panel of `pcaExplorer`
The figure shows the *Data Overview* panel of the `pcaExplorer` application, including a metadata table (**A**), a sample-to-sample heatmap (**B**), a summary of general information on the dataset (**C**) and a bar plot of the number of million reads of each sample (**D**). Additionally, the figure highlights the 'Group/color by' selector (**E**) in the sidebar panel, used to colour the visualisations by the condition of the samples.

After initial quality control, the *Samples View* panel displayed the Principle Component Analysis of the dataset (Figure 23). PCA is a dimensionality reduction technique that identifies the key patterns of variation in a dataset by transforming the data into a set of uncorrelated variables called Principal Components. When applied to gene expression data, PCA highlights the most significant sources of variation, allowing for an easy visualisation of sample relationships, detection of similar groups of samples, and identification of outliers. The panel showed a PCA plot of the first and second PCs, with each dot representing a sample (Figure 23A).

The PCA result of the Treg dataset showed a clear separation between the two condition groups, with the first PC explaining nearly 90% of the variance in the data and separating the samples based on condition. This indicated that the condition of the samples was the primary driver of variation, already hinting that this might also be a major factor influencing differential gene expression. Via the sidebar panel (Figure 23B), I set the number of most variable genes used for the PCA calculation to 500. This number balances the influence of biologically relevant variability, while minimising noise from consistently expressed genes; a choice also discussed and recommended in our manuscript [Ludt et al., 2022].

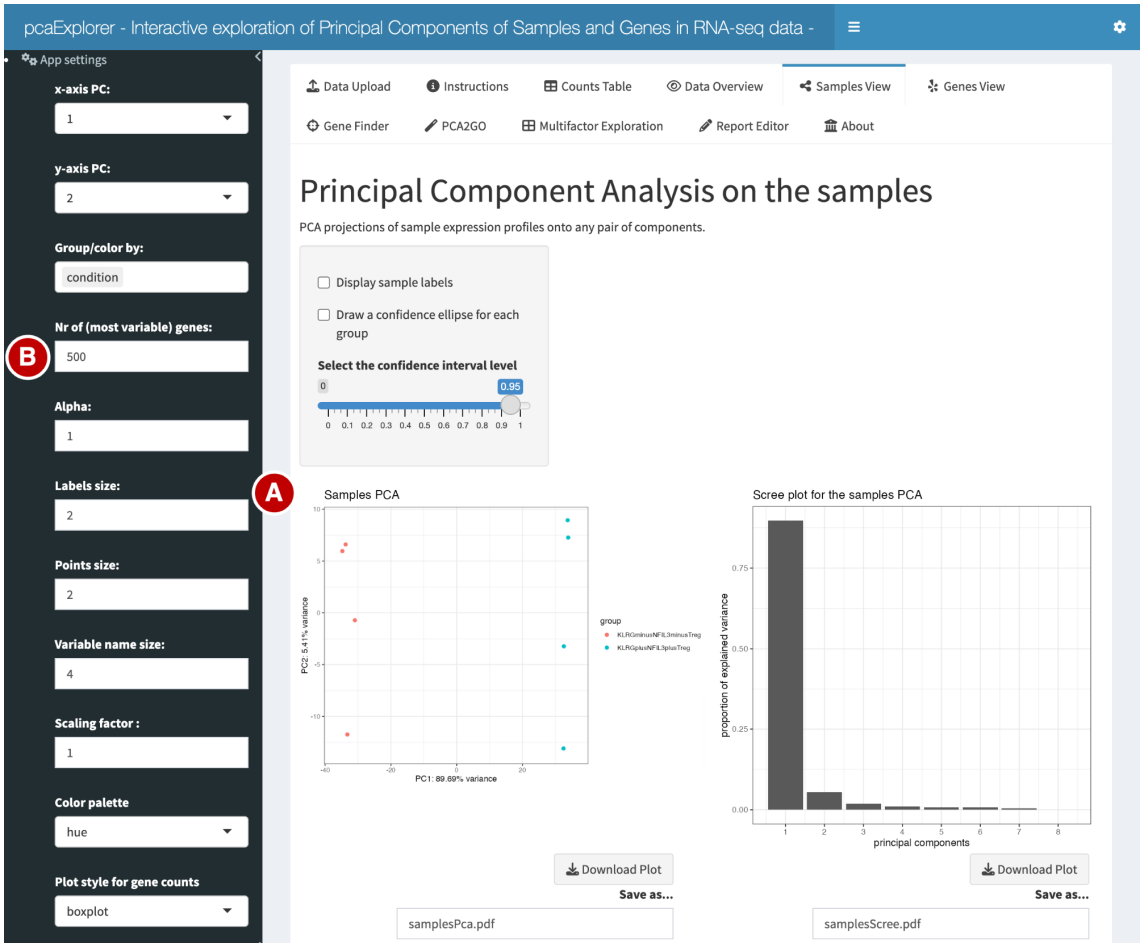


Figure 23 – The *Samples View* Panel of `pcaExplorer`
The figure shows the *Samples View* panel of the `pcaExplorer` application, highlighting the results of the PCA on the Treg dataset (**A**). In the PCA plot, the abscissa and ordinate show the first and second Principal Component, respectively, and the dots represent individual samples coloured by their respective condition. In the sidebar, the number of genes used for the PCA was set to 500 (**B**).

As a last step in this exploratory data analysis, the *Gene Finder* panel was used to visualise the expression of individual genes in the dataset. Using the text input field of this panel (Figure 24A), I selected a small subset of genes from the dataset (in Figure 25A Nfil3) to generate a box plot of their normalised expression counts on a log10 (logarithm to base 10) scale (Figure 24B). In order to facilitate the comparison of the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples, the box plot was coloured and divided by the condition of the samples.

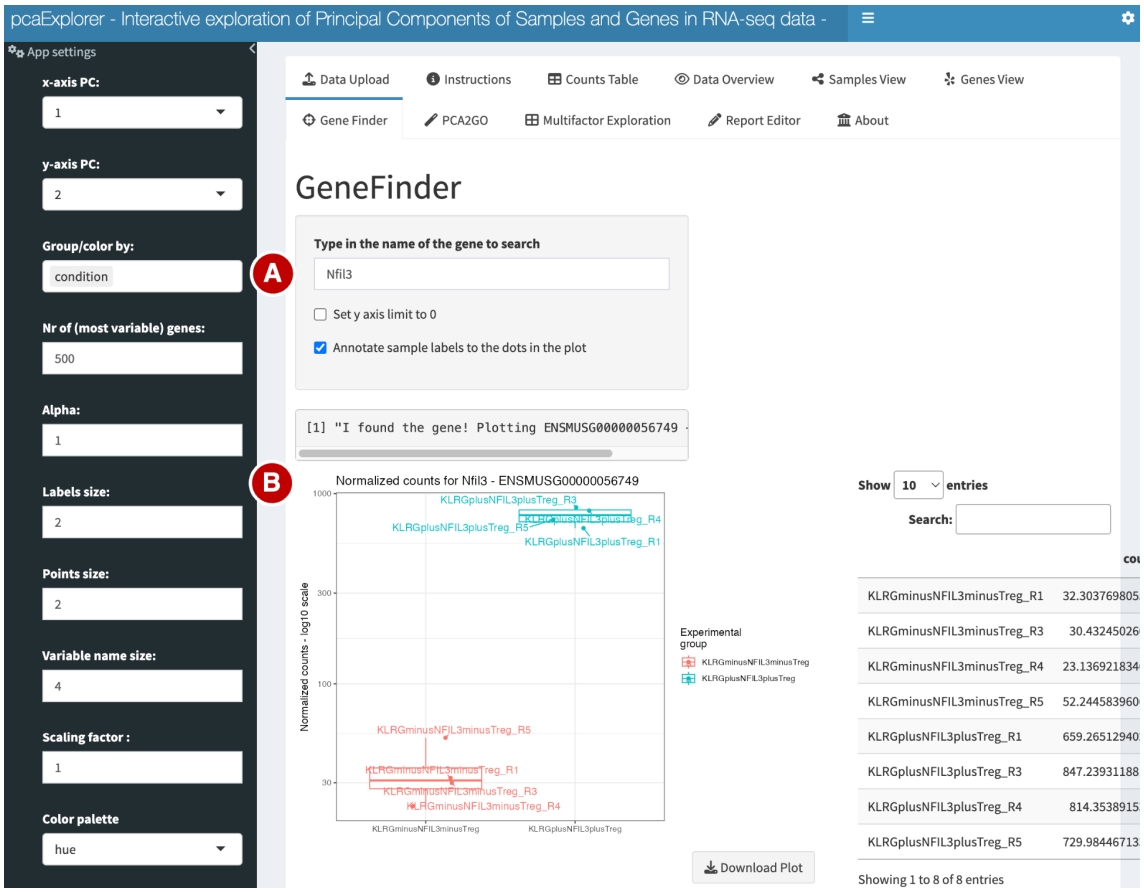


Figure 24 – The *Gene Finder* Panel of `pcaExplorer`
The figure shows the *Gene Finder* panel of the `pcaExplorer` application, in which specific genes from the data can be selected (**A**) to generate a box plot of the normalised expression counts of the genes (**B**). The box plot shows the expression of Nfil3 across the different conditions of the Treg dataset.

Based on discussions with the original authors of Delacher et al. [2020], I used the *Gene Finder* panel to examine the gene expression of Areg (Amphiregulin) and Il10 (Interleukin10) alongside Klrg1 and Nfil3 across the different sample conditions. The resulting box plots are presented in Figure 25, showing that all genes exhibit a higher expression in $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples (blue box) compared to the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ (red box). This aligns with expectations, as Klrg1 and Nfil3 were used to distinguish Treg subpopulations during data preparation in the original publication by Delacher et al. [2020]. Additionally, Areg and Il10 were included due to their known association with different Treg subpopulations [Delacher et al., 2017, 2020].

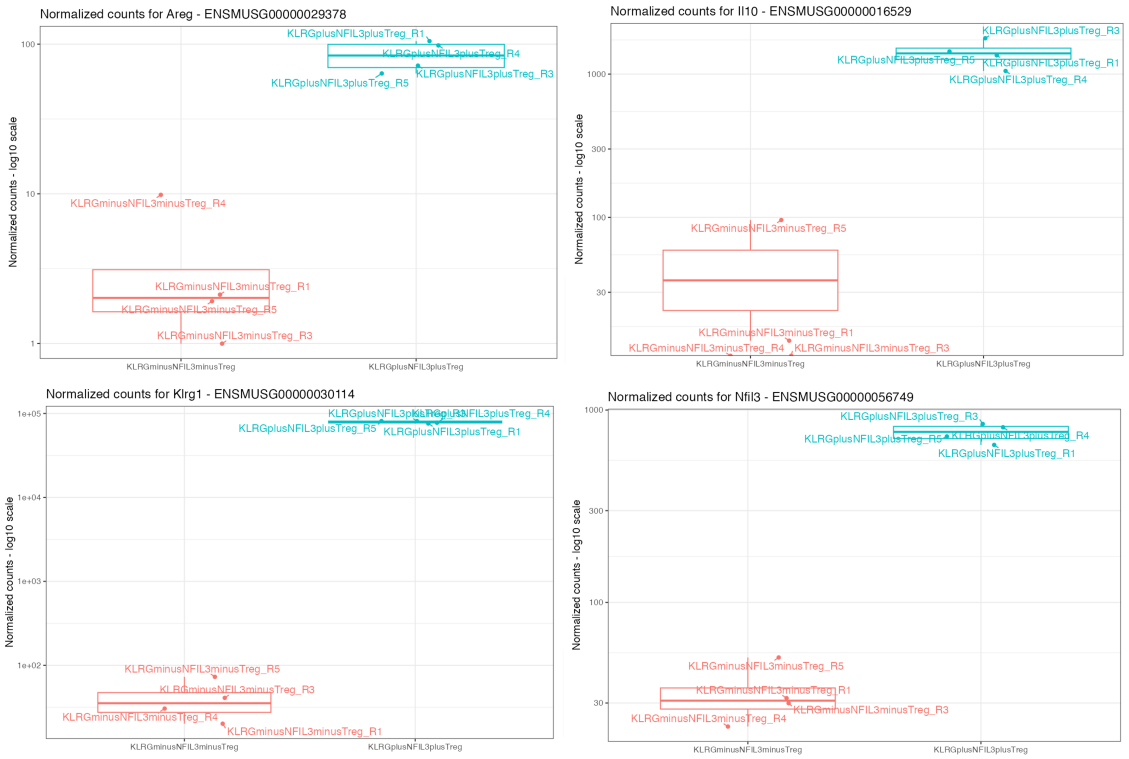


Figure 25 – Gene Expression Profiles
The figure displays box plots of the normalised counts for four genes. From left to right, top to bottom, the genes Areg, Il10, Klrg1 and Nfil3 can be seen. The box plots are split by the condition of the samples, which is also indicated by the colour. The ordinate visualises the normalised gene expression on a log10 scale. The figures were generated and downloaded from the *Gene Finder* panel of `pcaExplorer`.

Overall, the exploratory data analysis using `pcaExplorer` showed that the data used in this thesis was of high quality with no substantial outliers observable in the PCA plot. The conducted PCA also suggested that the sample conditions ($\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ vs. $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$) might be responsible for differential gene expression. This is in line with the original findings of the authors, who identified $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ Treg cells as precursors of tisTregST2 cells, with $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ representing a later stage, more closely resembling tisTregST2 gene expression. This was also further observed in the expression of individual genes in Figure 25, in which Areg and Il10, which were identified to be marker genes of tisTregST2 cells [Delacher et al., 2017, 2020], could be observed to have a higher expression in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ Treg samples.

In the next step of the analysis of the data, differentially expressed genes were identified using `ideal` [Marini et al., 2020], followed by an interpretation of the data using `GeneTonic` [Marini et al., 2021], as proposed in the original *Basic Protocol 2* and *3* of our publication [Ludt et al., 2022].

3.3.3 Differential Gene Expression Analysis using `ideal` and `GeneTonic`

Following the initial exploratory data analysis using `pcaExplorer`, the differentially expressed genes of the dataset were determined using the `ideal` [Marini et al., 2020] and `GeneTonic` [Marini et al., 2021] packages according to the workflow presented in out manuscript [Ludt et al., 2022]. In this step, the differentially expressed genes were first determined using `ideal` before the results were explored with `GeneTonic`.

In order to use `ideal`, the `DESeqDataSet` object and the annotation `data.frame` object from the previous section were used as input. Alternatively, for users with only little experience in R programming, `ideal` provides a step-by-step approach to generate the necessary input objects from the count and metadata. However, this step-by-step approach will not be discussed in this thesis; readers can find the details in our protocol manuscript [Ludt et al., 2022], especially Figure 13 and 14 of *Basic Protocol 2* document this data upload and generation process.

For this thesis, an `ideal` instance was launched using the `ideal()` function, with the `DESeqDataSet` object and the annotation `data.frame` object provided as parameters. This command opened an `ideal` session in the system's default web browser, which in this case was Google Chrome (version 129.0.6668.89), running on an Apple MacBook Pro with an Apple M1 Pro chip and macOS Ventura (version 13.2.1).

Upon launch, the `ideal` instance displayed the *Welcome* panel, which provided basic information about the package (Figure 26). The panel's header indicated the input status of the necessary workflow elements via coloured boxes: green for successfully loaded or generated data and red for missing data. From Figure 26A, it can be observed that the `DESeqDataSet`, containing the input gene expression data, as well as the annotation object were already available in the instance, while the differentially expressed genes were still pending generation. A summary of this information was also displayed in the sidebar's 'Quick Viewer' (Figure 26B).

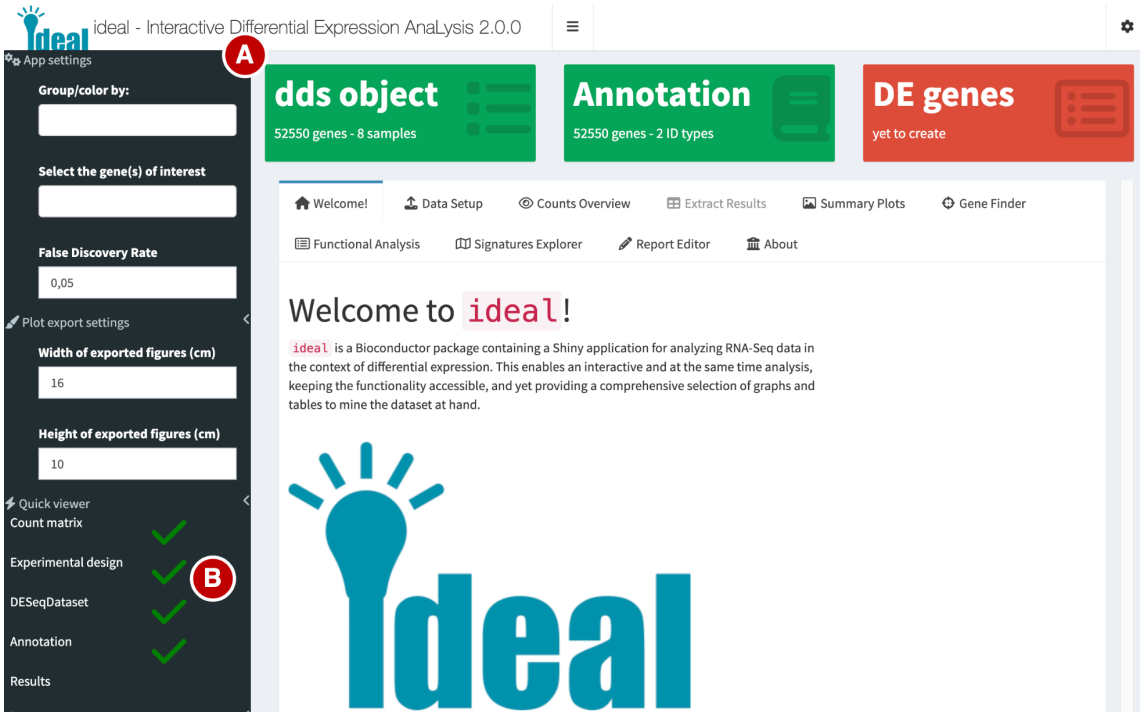


Figure 26 – The *Welcome* Panel of `ideal`
The figure shows the *Welcome* panel of the `ideal` application, with the header (**A**) displaying the status of the three essential workflow elements: the `DESeqDataSet` object (green, indicating that it is available), the annotation object (green, also available), and the differentially expressed genes (red, yet to be created). This information is also summarised in the 'Quick Viewer' in the sidebar (**B**).

Following the proposed workflow of *Basic Protocol 2*, the *Data Setup* panel was explored next (Figure 27). This panel can be used to set up the data from the count data and metadata similar to `pcaExplorer`. In the 'Step 1' box (Figure 27A), users can provide their data to the application, before they can generate the `DESeqDataSet` object in the 'Step 2' box (Figure 27B). The boxes will indicate if the data is already provided to the application, as shown in the figure.

In this thesis, the *Data Setup* panel was used to run the `DESeq()` function from the `DESeq2` package [Love et al., 2014], which is a usual step applied after providing the data to `ideal`. The function determines the differentially expressed genes in a given `DESeqDataSet` object, using a generalised linear model (GLM) to model the relationship between the normalised read counts and the experimental conditions of the data [Love et al., 2014]. Afterwards, the function applied the Wald test [Wald, 1943] to assess the significance of the coefficients associated with each condition in the GLM. After correction for multiple testing using the Benjamini-Hochberg procedure [Benjamini, Hochberg, 1995], the results are presented as a list of differentially expressed genes, each annotated with a log2 fold change, p-value and adjusted p-value [Love et al., 2014].

In `ideal`, the `DESeq()` function was executed using the 'Run DESeq!' button shown in Figure 27C. Once completed, `ideal` provided a summary of the results (Figure 27D), indicated how many of the overall number of genes with non-zero count are up- or downregulated. For the Treg dataset used in this thesis, the `DESeq()` function identified 5273 (14%) of the genes as significantly differentially upregulated, while 3531 (9.5%) of the genes were identified as downregulated. To minimise false positives, the False Discovery Rate (FDR) was used. The FDR represents the expected proportion of false positives among the results and was set via the sidebar (Figure 27E) to 5%. This is a standard cutoff commonly applied in differential expression analyses between conditions [Robinson et al., 2010; Love et al., 2014; Ludt et al., 2022; Chen et al., 2024].

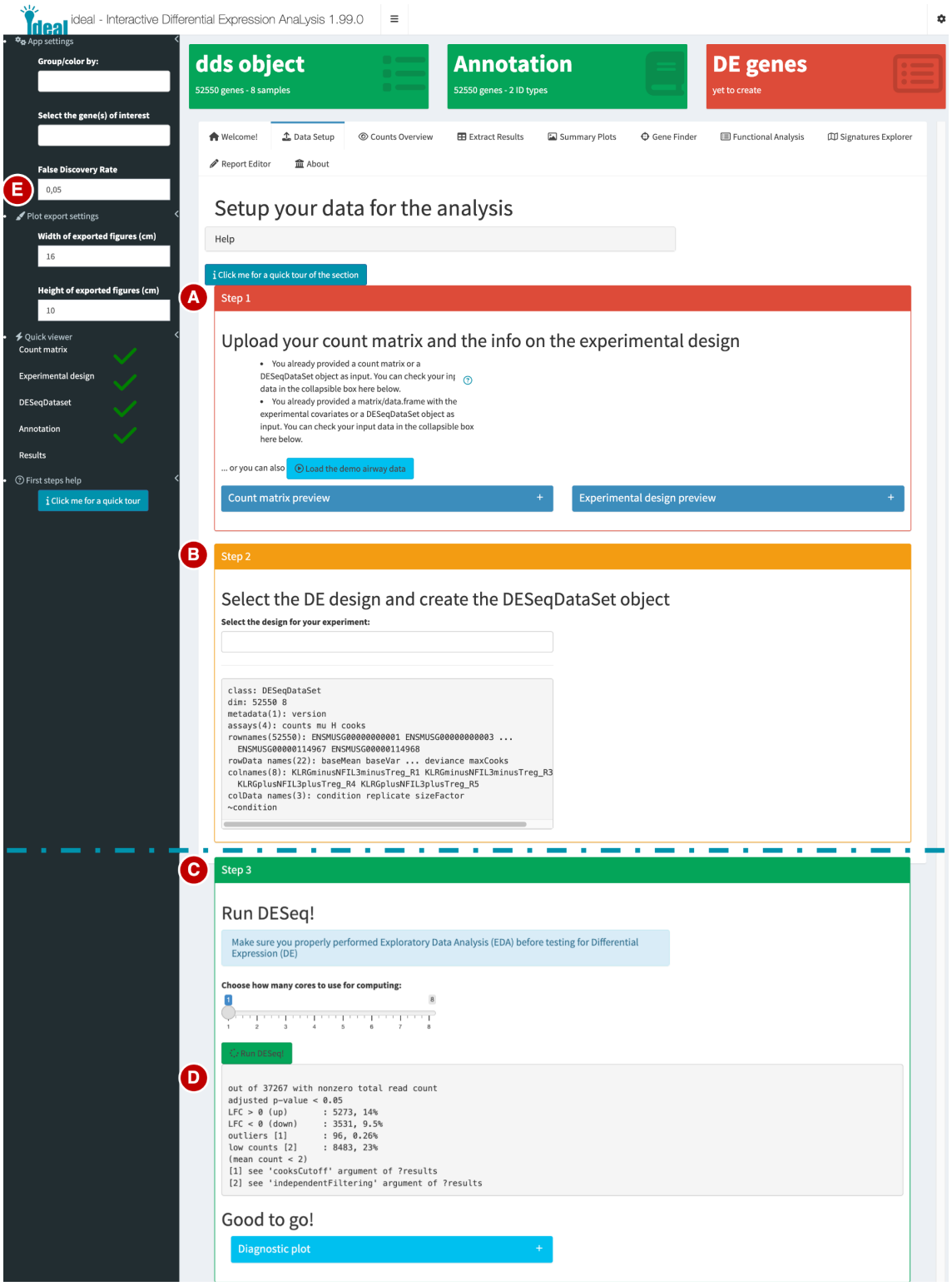


Figure 27 – The *Data Setup* Panel of `ideal`

The figure shows the *Data Setup* panel of the `ideal` application, which can be used to provide the data to the application (**A**) and generate the `DESeqDataSet` object (**B**). Once the data is provided, differentially expressed genes can be calculated using the 'Run DESeq!' button in the 'Step 3' box (**C**). Once calculated, the results will be summarised in the grey box below the button (**D**). Via the sidebar menu, the False Discovery Rate can be set (**E**). In this figure, parts of the original panel have been omitted to enhance clarity. This is indicated by the dotted, blue line.

After identifying differentially expressed genes, these genes were explored in the *Extract Results* panel of `ideal` (Figure 28). In this panel, I first chose the experimental factor on which the contrast is build upon. This contrast defined the characteristic by which the samples were grouped and their gene expression compared. For the Treg dataset, this was set to the samples' condition, as seen in Figure 28A.

Once the contrast was set, the numerator and denominator level for the fold change calculation had to be defined. In the analysis of bulk RNA-seq data, the numerator is typically set to the treatment group of the experiment, while the denominator is represented by the control group, establishing this group as the baseline of the gene expression. In the presented analysis, the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ samples were chosen as the denominator and the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples as the numerator (Figure 28A).

The resulting fold change then represented the ratio of gene expression between the two conditions, i.e., how much the gene expression changed between the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ and $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples. In this analysis, a FC of greater than 1 indicated that a gene was upregulated (i.e., more highly expressed) in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ (the numerator) samples compared to the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ (control group/denominator) samples. Conversely, a fold change of less than 1 indicated a downregulated gene.

Once the contrast was defined, I used the 'Extract the results!' button (Figure 28B) to extract the differentially expressed genes and generate a table of these genes. The summary indicated again, that 14% of the expressed genes were upregulated, while roughly 10% were downregulated. In the generated table of differentially expressed genes (Figure 28C), it can be observed that Ctla4 (cytotoxic T-lymphocyte-associated protein 4) and Ccr7 (C-C motif chemokine receptor 7) were the "top" results, with the smallest p-value. From the table it is also evident, that Ctla4 was upregulated with a positive fold change, while Ccr7 was downregulated.

In a second step, the table was sorted by the log2 fold change (log2FC), which is the fold change scaled by the logarithm to base 2 (log2). This sorting revealed Klrg1 as the gene with the highest positive log2FC of nearly 10, indicating that it is $2^{10} = 1024$ times more expressed in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples compared to the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ samples. While this difference was already observable with `pcaExplorer` (see Figure 25), the analysis with `ideal` revealed that the gene was also significantly differentially expressed between the samples of the two conditions; a result which was also found by the original authors [Delacher et al., 2020]. Other genes with a high, positive log2FC included Ccd2a (coiled-coil and C2 domain containing 2A), Ccr10 (C-C motif chemokine receptor 10) and Tigit (T cell immunoreceptor with Ig and ITIM domains).

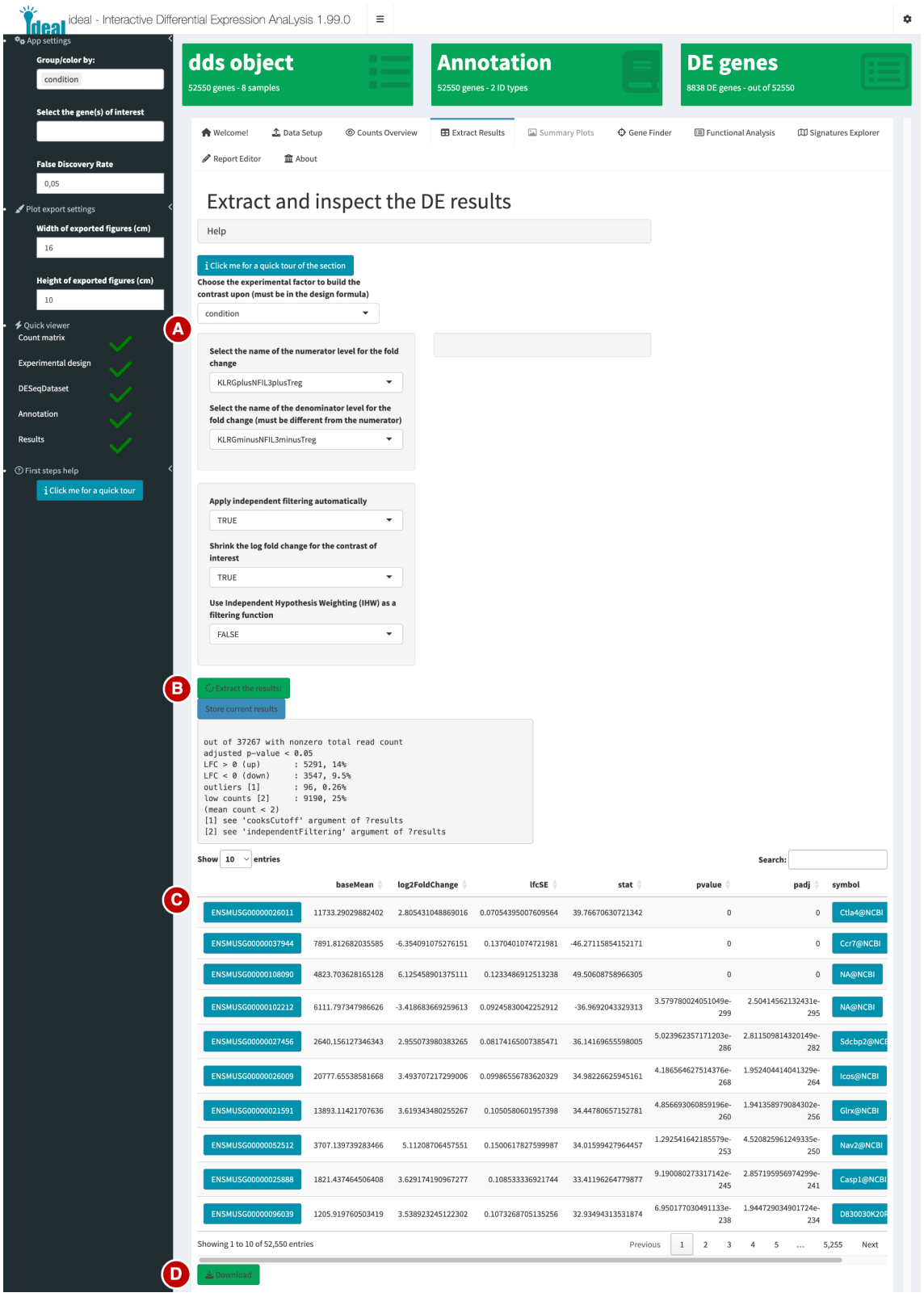


Figure 28 – The *Extract Results* Panel of `ideal`
The figure shows the *Extract Results* panel of the `ideal` application, which can be used to first set the experimental factor for the contrast (**A**), as well as the numerator and denominator of this contrast. Once the contrast is set, the 'Extract the results!' button (**B**) can be used to generate a table of the differentially expressed genes (**C**). This table can also be downloaded for further inspection using the 'Download' button (**D**).

In a last step, a functional enrichment analysis was performed with `ideal` using the *Functional Analysis* panel (Figure 29). In this panel, I first specified the GO category of interest as "Biological Process" (BP) (Figure 29A). The three available categories or ontologies of the GO database are "Biological Process" (BP), "Molecular Function" (MF) or "Cellular Component" (CC), which all describe different aspects of a gene product's role within a biological system. The BP ontology includes larger biological processes or programs which are generally accomplished by multiple molecular activities, while the MF ontology describe activities at the molecular-level. In contrast, the CC ontology indicates where the gene product is active within the cell [Ashburner et al., 2000; Carbon et al., 2019]. For the Treg dataset used in this thesis, the BP ontology was chosen to analyse the larger biological processes between the two subpopulations of Treg cells.

After selecting BP as the GO ontology, I defined that the functional enrichment analysis should be calculated on all DEGs (Figure 29B), before starting the computation of the analysis using the 'Perform gene set enrichment analysis on the up- and down-regulated genes - topGO' button (Figure 29C), which used the `topGO` package [Alexa et al., 2006] to perform an Over-Representation Analysis on the DEGs. Once calculated, the results of the analysis were displayed in a table (Figure 29D). This table, sorted by p-value, indicated that the top differentially regulated pathways for the Treg dataset were "cell division" (GO:0051301) as well as "positive/negative regulation of apoptotic process" (GO:0043065, GO:0043066) and "positive regulation of T cell cytokine production" (GO:0002726).

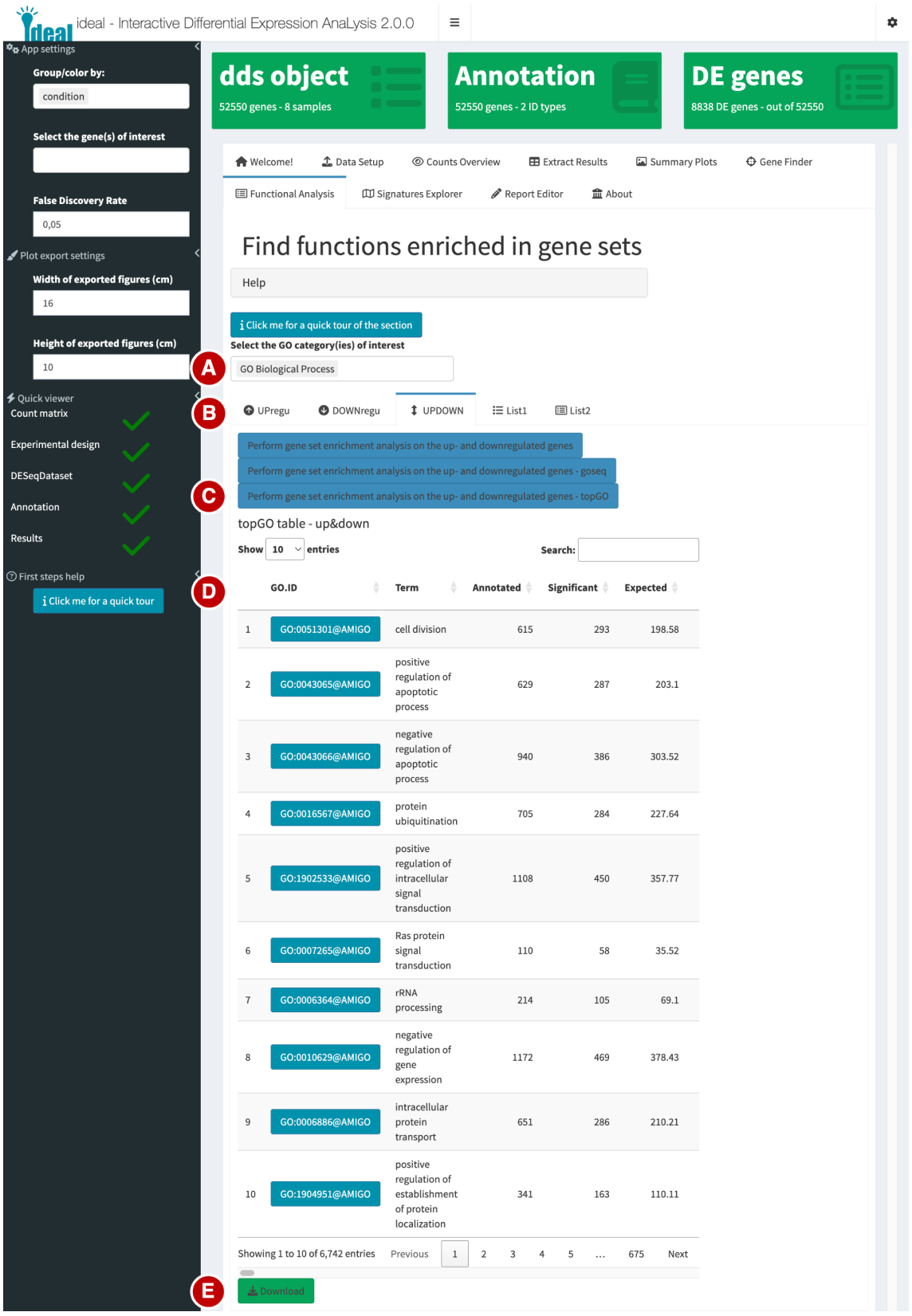


Figure 29 – The *Functional Analysis* Panel of `ideal`
The figure shows the *Functional Analysis* panel of the `ideal` application, which can be used to perform functional enrichment analysis on the DEGs by first determining the GO category of interest (**A**) and the list of DEGs on which the enrichment should be performed (either only up-, only downregulated, all DEGs, or two custom lists, **B**). The functional enrichment results can then be calculated using the available buttons (**C**) and will afterwards be displayed in a table (**D**). This table can also be downloaded for further inspection using the 'Download' button (**E**).

Following the data analysis with `ideal`, the DE and functional enrichment results were further interpreted using `GeneTonic` [Marini et al., 2021], following the standardised workflow of *Basic Protocol 3* of our manuscript [Ludt et al., 2022]. In order to use `GeneTonic`, the results were downloaded during the exploration of the data with `ideal`, specifically in the *Extract Results* and *Functional Analysis* panels (Figure 28D and Figure 29E). Since this provided the results in form of a CSV (comma-separated values) file, preprocessing was required, as these file formats could not be directly used `GeneTonic`. For this, the CSV files were first read into an active R session using the `read.csv()` function from the `utils` package [R Core Team, 2024]. The DEGs were then transformed to `DESeqResults` object using the `DESeqResults()` function from the `DESeq2` package [Love et al., 2014].

The functional enrichment results, on the other hand, were first filtered to the top 500 gene sets with the smallest p-value. This filtering was applied to streamline downstream analyses by reducing runtime and enhancing clarity through limiting the gene sets to a more manageable number. Afterwards, the filtered data was further prepared using the `shake_topGOtableResult()` and `get_aggrscores()` functions of the `GeneTonic` package [Marini et al., 2021]. The latter function is especially useful in the interpretation of the results, as it adds Z-scores to the data, which are a measure to quantify the general direction of regulation of a gene set [Marini et al., 2021].

Afterwards, the prepared DEGs and functional enrichment results, together with the original count data `DESeqDataSet` object and the annotation `data.frame`, were assembled to a `GeneTonicList` object following the instructions of our protocols manuscript [Ludt et al., 2022]. Using this `GeneTonicList` object, an instance of the app was launched using the `GeneTonic()` function. This launched an instance of the app showing the *Welcome* panel. The *Welcome* panel, similar to the panels in the `pcaExplorer` and `ideal` applications, served as entry point to the exploration and analysis by providing expandable elements, showing tables of the input data (Figure 30A), as well as summary boxes on the data (Figure 30B). From these boxes, several characteristics of the Treg dataset could be observed directly, such as the number of genes in the original count data (52,550) or the number of differentially expressed genes (8,838).

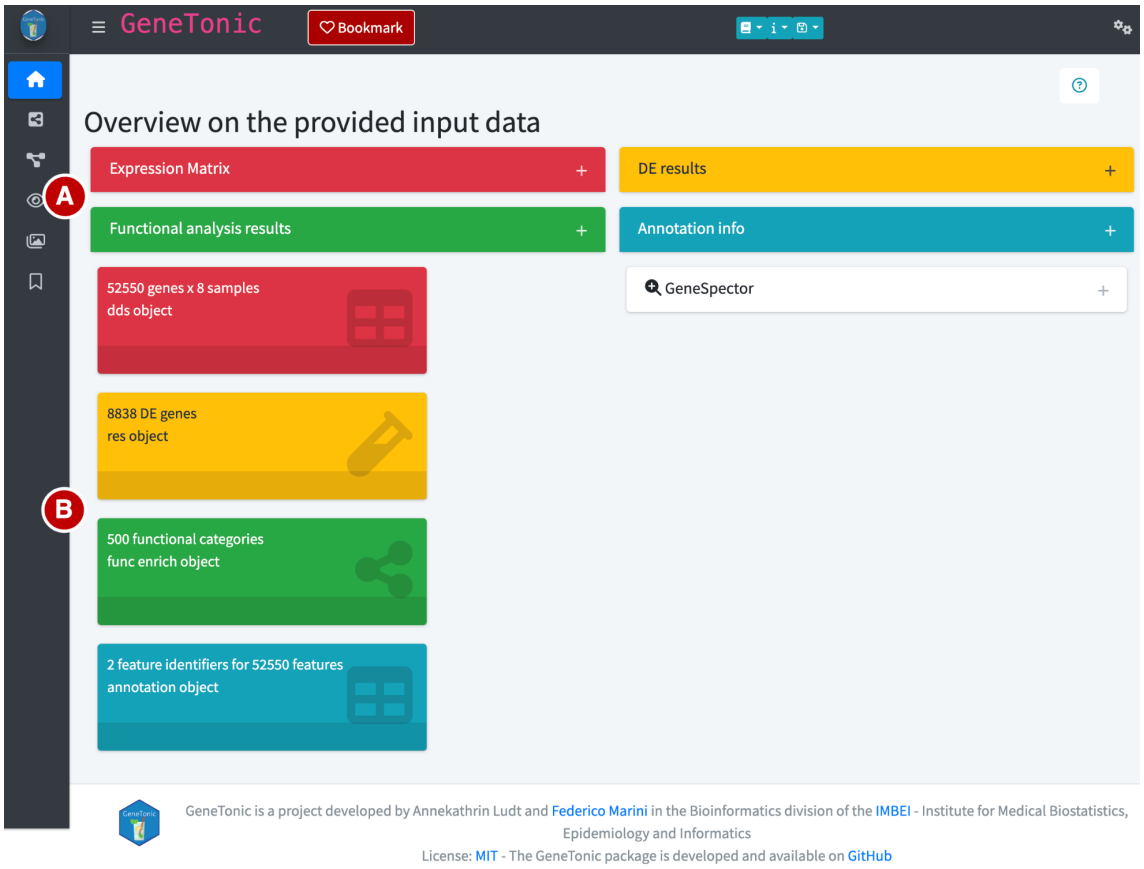


Figure 30 – The *Welcome* Panel of `GeneTonic`
The figures shows the *Welcome* panel of the `GeneTonic` application, which provides an overview of the input data in collapsible boxes (**A**) and summary boxes (**B**).

Following *Basic Protocol 3* of our manuscript, I explored the *Gene-Geneset* panel next. This panel featured an interactive 'Gene-Geneset' graph, displaying the top 15 most significantly regulated pathways (based on smallest p-values) and their associated genes (Figure 31). In this graph, oval nodes represented genes, rectangular nodes represented gene sets, and edges connected genes to the gene sets they belong to. The graph was highly interactive, allowing the selection of individual gene sets or genes by either clicking on the respective node or through the selection box in the upper left corner.

In the 'Gene-Geneset' graph, I selected the "positive regulation of T cell cytokine production" pathway (Figure 31A). Upon selection, additional visualisations appeared in the 'Geneset Box', including a heatmap and a volcano plot (Figure 31B). The heatmap showed the expression of the member genes of the selected gene set, revealing two distinct groups of genes with alternating expression patterns: some were upregulated (red) in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples, while others showed higher expression in the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$ samples.

In the volcano plot, the log2FoldChange of all genes was plotted against the negative, log10-scaled p-value, with the member genes of the "positive regulation of T cell cytokine production" pathway coloured and annotated. This also showed that there was no clear pattern observable in the expression of the gene set members, with some genes having a positive FC (i.e., upregulation in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples) and some having a negative FC. Besides the two visualisations, the 'Geneset Box' also provided additional

information about the gene set, such as the GO ID, a definition and alternative names of the pathway.

Besides gene sets, also individual genes were be selected in the 'Gene-Geneset' graph, which lead to additional visualisations of the selected gene shown in the 'Gene Box' (Figure 31C). This included a box plot of the gene expression and links to external databases for deeper exploration. In Figure 31C, Nfil3 was selected, showing its higher expression in the $\mathrm{Klrg1}^{+}\mathrm{Nfil3}^{+}$ samples compared to the $\mathrm{Klrg1}^{-}\mathrm{Nfil3}^{-}$; a pattern previously observed with `pcaExplorer` in Figure 25. Additionally, the 'Gene Box' confirmed that Nfil3 was significantly differentially expressed in the Treg dataset.

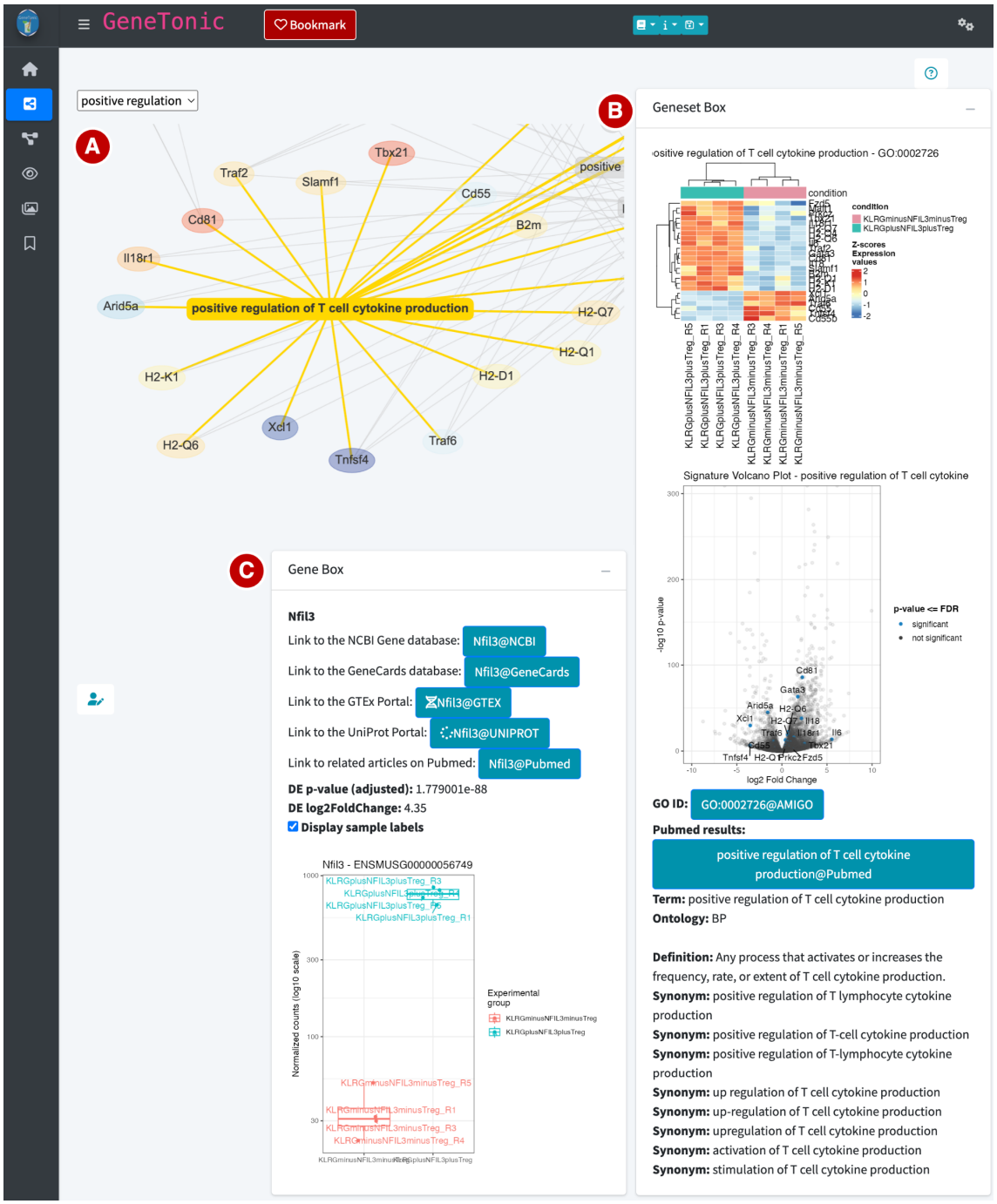


Figure 31 – The *Gene-Geneset* Panel of `GeneTonic`
The figure shows the *Gene-Geneset* panel of the `GeneTonic` application, which visualises the relationships between genes and gene sets in the 'Gene-Geneset' graph (**A**). In this graph, genes are depicted as oval nodes, while gene sets are shown as rectangular nodes, and edges are drawn between gene sets and their respective member genes. Upon selection of individual gene sets or genes, additional visualisations as well as information are shown in the 'Geneset Box' (**B**) and 'Gene Box'(**C**), respectively.

In a last step, I explored the results of the *Enrichment Map* panel. This panel showed an enrichment map of the 15 most regulated pathways (i.e., those with the smallest p-values) based on the functional enrichment results. The individual nodes of the enrichment map represented gene sets, with the node size reflecting the number of DEGs associated with each gene set (Figure 32A). Additionally, edges were drawn between nodes which had a high overlap of DEGs, while the colour of the nodes was chosen to represent the p-value of the pathways.

Upon selection of an individual node, such as "positive selection of T cell cytokine production", a heatmap was displayed in the 'Geneset Box' (Figure 32B). This box was similar to the 'Geneset Box' of the *Gene-Geneset* panel (see Figure 31B), with the heatmap also depicting the gene expression of the member genes of the selected gene set. Similar to its counterpart in the previous panel, this 'Geneset Box' also provided additional information on the selected gene set, such as the GO ID (GO:0002726), p-value ($5.3e^{-06}$) and a short definition as well as alternative names of the pathway (Figure 32B).

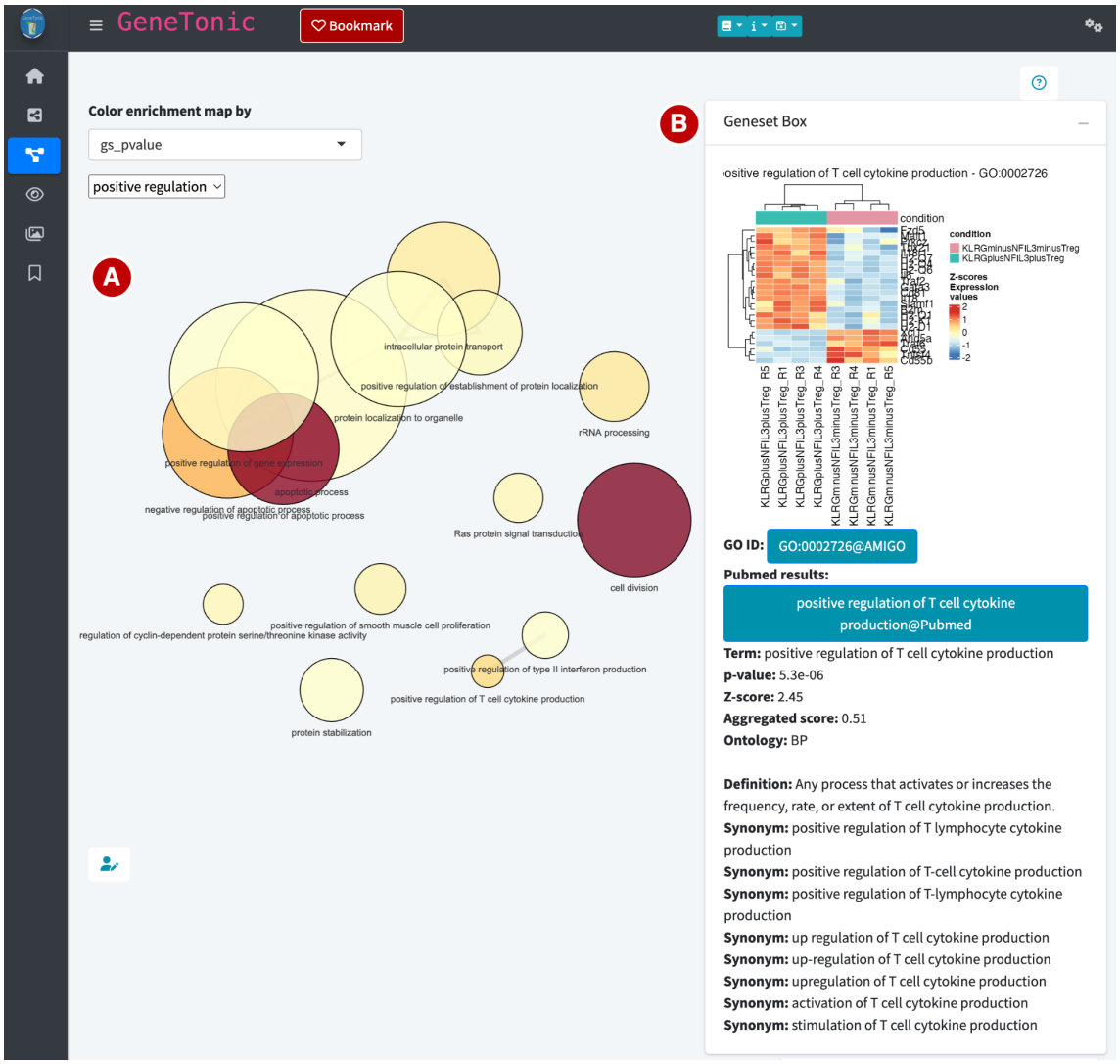


Figure 32 – The *Enrichment Map* Panel of `GeneTonic`
The figure shows the *Enrichment Map* panel of the `GeneTonic` application, displaying an interactive enrichment map of the top regulated pathways based on functional enrichment results (**A**). In the shown enrichment map, nodes represent individual gene sets, edges are drawn between gene sets with a high overlap and the size of the nodes represents the number of DEGs in each pathway. The nodes were coloured by p-value using the input selector in the upper left corner. Upon selection of a node, a gene expression heatmap is shown in the 'Geneset Box' (**B**).

The data analysis using GeneTonic provided some initial insights into the functional enrichment results and the relationships between the individual gene sets in the result list. In order to further investigate these relationships and explore the results, GeDi was used in the next step. GeDi can provide deeper insights into the relationships between the individual gene sets in the functional enrichment results and help uncover larger patterns and molecular mechanisms in the data, which is not directly possible with GeneTonic.

3.3.4 Functional Enrichment Result Interpretation using GeDi

The literature review results in Section 3.2 showed that the majority of reviewed articles usually completed their functional enrichment analysis by selecting and highlighting the top gene sets with the smallest p-value. Only a small subset of articles aggregated their results to gain an overview of the biological patterns included or focused on unexpected findings. Using GeDi, the exploration and interpretation of functional enrichment results can be greatly simplified and new hypotheses can be generated from the data, as will be shown in this section.

In order to start the exploration of the data with GeDi, the GeneTonicList object generated in Section 3.3.3 was used. While a GeneTonic list object contains additional information beyond what is needed for a successful exploration with GeDi, the package was intentionally designed to include the list object among the accepted input file formats. This choice was made to ensure that the GeDi package could be seamlessly integrated into the standardised bulk RNA-seq workflow that we presented in our article [Ludt et al., 2022]. In order to ensure this, GeDi internally extracts the functional enrichment results from the GeneTonicList object without further need for data preparation by the user.

In this thesis, an instance of GeDi was launched using the **GeDi()** function, providing the data upon the call of the function with the **gtl** argument (short for **G**ene**T**onic**L**ist). This opened a running instance of GeDi, showing the *Welcome* panel of the Shiny application (Figure 33).

Similar to the previously presented applications, the *Welcome* panel of GeDi also serves as entry point, providing an overview of the features and functionality of the app. This design was intended to not only align with the user experience of other packages developed by our group, but also to lower the entry barrier for users. By providing a familiar interface, it helps ease the learning curve, especially for those who are new to our tools or Shiny applications in general.

In a first step in the *Welcome* panel, I started the interactive tour of the panel using the 'Help' button in the upper right corner (Figure 33A). This opened a modal window displaying an interactive tour, which guided through the content of the panel (Figure 33B). This modal window highlighted and explained individual components of the application, helping to understand how to use them effectively. Once the tour was completed, I used the sidebar panel to navigate to the other panels of GeDi (Figure 33C).

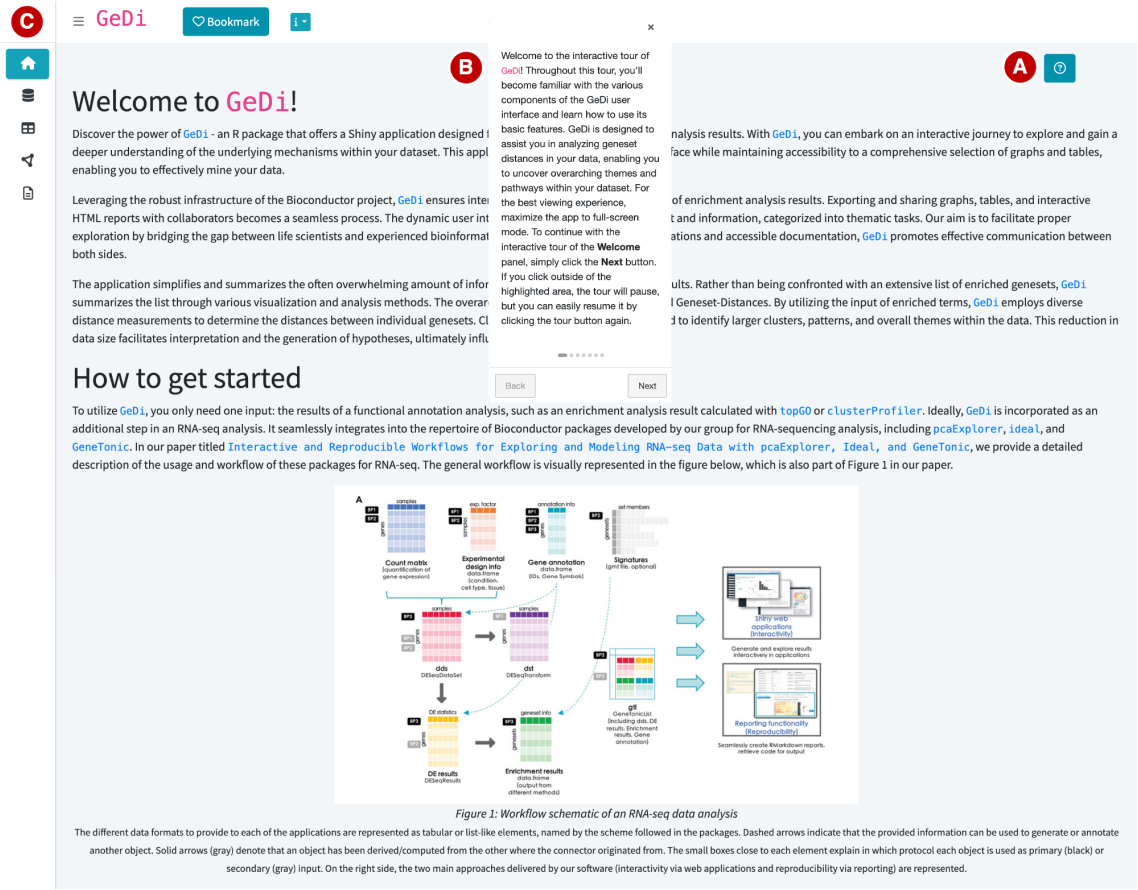


Figure 33 – The *Welcome* Panel of `GeDi`
The figure shows the *Welcome* panel of the `GeDi` application, which provides descriptions about the package's functionality and the application's interactive elements, such as the 'Help' button (**A**). Clicking the 'Help' button will open up a modal window (**B**), explaining the individual parts of the panel, such as the sidebar panel (**C**) and other elements. It should be noted that the tour's modal window is not directly presented in this manner within the app but was rather rearranged during figure assembly to improve clarity.

In the next step of the data exploration, I navigated to the *Data Input* panel of `GeDi`, which can be used to provide the data to a running instance of the application, usually via the 'Browse' button in the 'Step 1' box (Figure 34A). As the data was already provided upon the call of the `GeDi` function in the presented showcase, the 'Genesets preview' box was used to inspect the provided data (Figure 34B). This box presented the input data in form of a table, enabling an quick overview of the data. From this table, it is also apparent that the pathways with the smallest p-value were "cell division" and "positive/negative regulation of apoptotic process" - results, which were already observable in `ideal` (see Figure 29).

Afterwards, I used the 'Optional Filtering Step' box to remove large and possibly redundant gene sets from the data (Figure 34C). The histogram within the box showed the distribution of gene set sizes in the functional enrichment results of the Treg dataset, ranging from small numbers close to 0 to over 6000. Using the zoom feature available in the panel (Figure 34D), I closely examined the larger sets, revealing "biological_process" as the largest set, associated with 6,140 of the 8,838 DEGs. As previously discussed in Section 2.4.1, these extensive and generic sets can be detrimental to downstream analysis steps by prolonging the runtime.

During the exploration of the Treg dataset, it was also expected that these gene sets have a second detrimental effect besides their effect on the runtime: With nearly 70% of the differentially expressed genes involved in the pathway, "biological_process", but also other large gene sets, were expected to become a highly connected nodes (i.e., hub nodes) in the resulting clustering graph of the data due to the large overlap in genes with other gene sets. As such, these hub nodes could have overshadowed other, meaningful clusters in the data and lead to cluttered graphs and visualisations. Hence, it was decided to filter large and generic gene sets with a size of at least 200 genes from the data (Figure 34E). This removed 79 of the original 500 gene sets from the data.

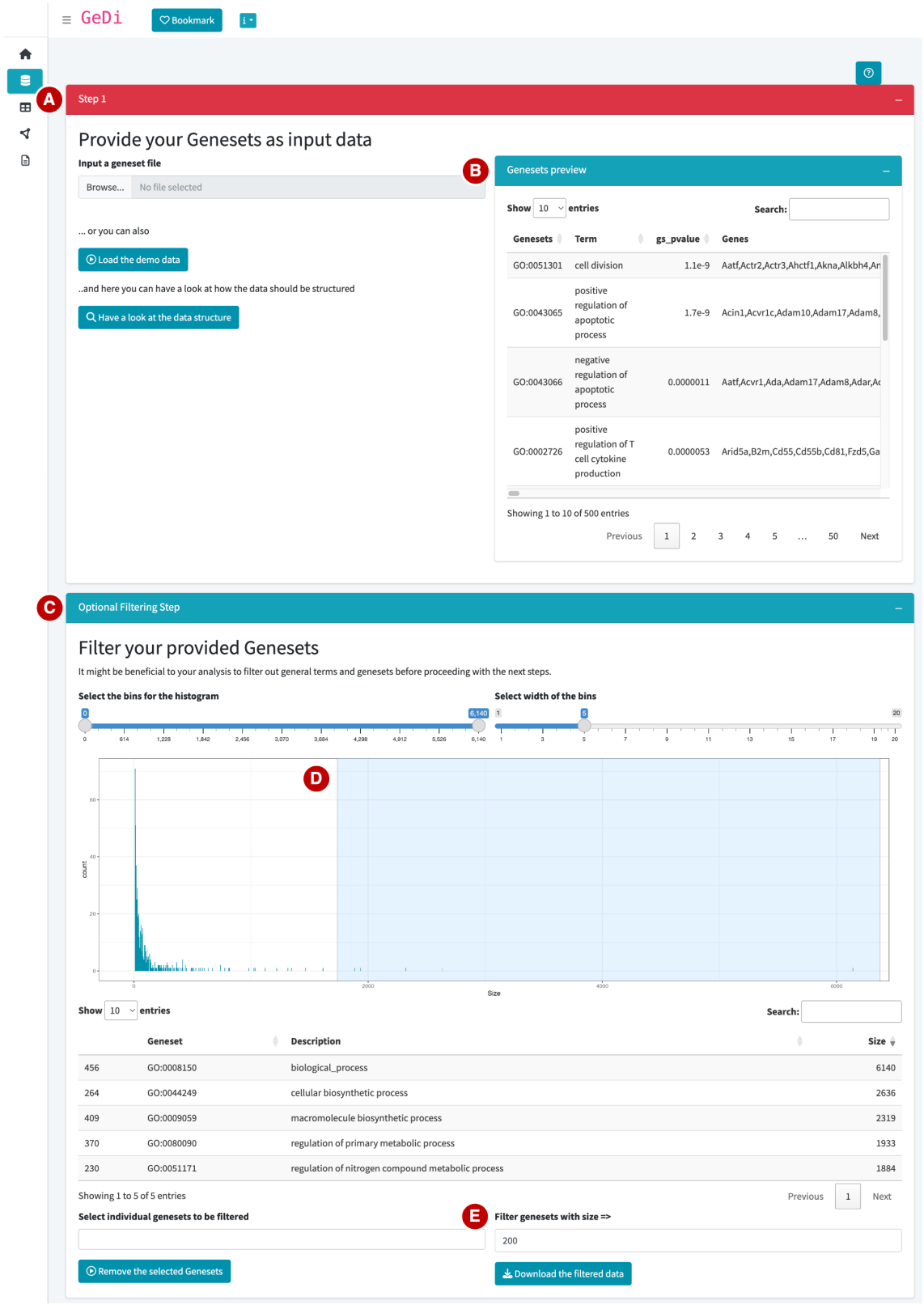


Figure 34 – The Upper Half of the *Data Input* Panel of `GeDi`
The figure shows the upper half of the *Data Input* panel of `GeDi`, used to provide the data to the application in the 'Step 1' box (**A**). The data can afterwards be observed in the collapsible 'Genesets preview' box (**B**). The 'Optional Filtering Step' box (**C**) visualises the gene set sizes as a histogram, which provides an interactive zooming feature (**D**) and the possibility to filter gene sets based on their individual IDs or a size threshold (**E**).

Once the data was filtered, the 'Step 2' box was used to select the species of the data, using the input selector (Figure 35A). This selection triggered the appearance of the 'Step 3' box (Figure 35B), which was used to download the protein-protein interaction information from the STRING database via the 'Download PPI matrix' button. The download of the PPI data to the local machine concluded the data preparation steps available in the *Data Input* panel (Figure 35C). Using the sidebar, the exploration was continued in the *Distance Scores* panel.

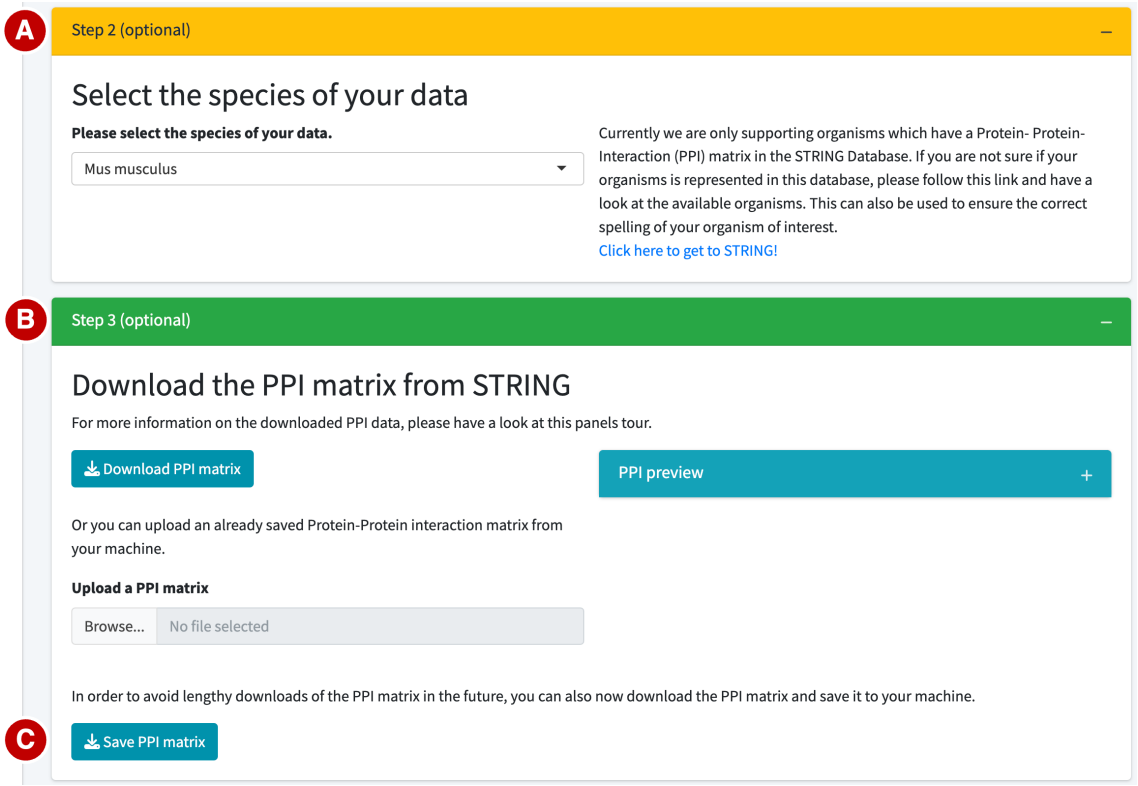


Figure 35 – The Lower Half of the *Data Input* Panel of `GeDi`
The figure shows the lower half of the *Data Input* panel of `GeDi`, used to specify the species of the data in the 'Step 2' box (**A**), before downloading the PPI data in the 'Step 3' box (**B**). The downloaded PPI data can also be saved to the local machine using the 'Save PPI matrix' button (**C**).

3.3.4.1 Quantification of Gene Set Dissimilarity using Distance Scores

The primary goal of `GeDi` is to summarise a large number of gene sets into individual, more specific groupings which highlight different biological patterns and pathways in the data. This is achieved in two steps: (1) the (dis)similarity of each pair of gene sets is determined by calculating the distances of each pair; (2) the gene sets are aggregated into clusters, based on their distance scores.

These distances are calculated in the *Distance Scores* panel (Figure 36), where the individual available distance metrics can be selected in the 'Calculate Distance Scores for your Genesets' box (Figure 36A). During the preparation of this thesis, I evaluated all of the available distance metrics and compared the resulting distances. In the end, I decided to highlight the results of the pMM and the GO semantic distance score in this thesis, as these scores efficiently aggregated the gene sets into medium-sized, compelling clusters. This observation was also supported by our collaboration partners and

the original authors Delacher et al. [2020] of the Treg datasest used in this thesis (see Section 3.3.1), as the pMM and GO semantic distance score generated promising results facilitating the interpretation of the data and highlighting new aspects, which were not explored in the original manuscript of Delacher et al. [2020] and will be discussed later in this thesis.

Hence, to calculate the distances, I first selected the pMM score from the available scores and set the value of α to 1, in order to balance the influence of the set-based and PPI-based components of the score (Figure 36B). Using the 'Compute the distances between the gene sets' button, the distances were calculated (Figure 36C).

Once calculated, the visualisations of the distances could be observed in the 'Geneset Distances Scores' box (Figure 36D). The box provides the option to select the distance metric used for the visualisations from a drop-down menu as well as the possibility to download the calculated distances (Figure 36E), which was used in this thesis to compare the results of the different distance metrics and reduce the runtime during the several, iterative explorations of the Treg dataset.

In Figure 36F and G, heatmap visualisations of the calculated pMM and GO semantic distances are displayed. Overall, the two metrics produced similar results, with most gene sets showing large distances close to 1, indicated by the predominant red colour. Only upon closer inspection of both heatmaps, similar gene sets could be observed. For the pMM distances (Figure 36F), these similar gene sets appeared around the diagonal, especially in the upper left and lower right corner. Similarly, the GO distance score results (Figure 36G) also showed gene sets with smaller distances around the diagonal. For this score, already some larger, potential clusters could be observed by the white and light-blue coloured entries around the diagonal.

Figure 36 – The *Distance Scores* Panel of GeDi
The figure shows the *Distance Scores* panel of GeDi, with the 'Calculate Distance Scores for your Genesets' box, which can be used to select one of the available distance metrics (**A**), set the α parameter for the pMM calculation (**B**), and compute the distances (**C**). Results are displayed in the 'Geneset Distance Scores' box (**D**), where a specific distance metric can be chosen for visualisation and calculated distances can be downloaded (**E**). The box also includes heatmaps of the pMM (**F**) and GO distances (**G**), arranged side-by-side here for enhanced clarity, although GeDi does not support this layout directly in the application.

Besides the comparison of the individual distance metrics, this thesis also evaluated the influence of the α parameter on the resulting pMM distances. α is a scaling factor in the pMM score which controls how strongly the PPI information influences the resulting distances (see Section 2.2.4 for the definition of the score).

In order to evaluate this effect of the protein-protein interactions on the resulting distances of the Treg dataset, various values of α were tested and compared (Figure 37). For this comparison, the functional enrichment results were further filtered to include only gene sets which had a size of less than 10 differentially expressed genes. This choice and filtering was applied to enhance the influence of the PPI component of the pMM score.

Naturally, with increasing size of the gene sets, the likelihood of shared genes increases. However, in this evaluation I wanted to draw specific focus to the α parameter and its regulation of the PPI component. Hence, I wanted to balance the size of the gene sets to ensure that the number of shared genes is small, but also not too small, as this would consequently reduce the likelihood of protein-protein interactions between the genes of two gene sets. A further reason for this filtering choice was the number of remaining gene sets. In the Treg dataset used in this thesis, 32 gene sets had a size smaller than 10, a number which can still be easily observed and interpreted in the resulting heatmaps of the distance scores.

Figure 37 shows the heatmaps of the resulting pMM distances using α values of 0, 0.5 and 1 (hereafter referred to as pMM_0, $\mathrm{pMM}_{0.5}$ and pMM_1) as well as the Meet-Min distances. According to the definition of the pMM score, at an α value of 0, the score is identical to the Meet-Min distance score. This can also be observed in the upper row of Figure 37, where the MM distances are shown on the left and the pMM_0 distances on the right.

For larger values of the scaling factor, it can be observed that the resulting distances start to differ from the MM results as the influence of the protein-protein interactions on the scores increased. However, overall the heatmaps remained similar. It seems that gene sets which already had an average distance score of around 0.5 (indicated by the white/light-blue colour) further decreased their distances through the additional PPI information. This can also be observed in the lower right corner of each heatmap or in the highlighted zoomed parts of the heatmaps.

Here, it can be observed that for the gene sets in this area, the distance score becomes gradually smaller with increasing values of α. In the initial MM and pMM_0 results, these gene sets had an average distance indicated by the white colour, which becomes increasingly more blue in the $\mathrm{pMM}_{0.5}$ and pMM_1 results. Additionally, Figure 37 highlights a part of the heatmaps as zoomed in and overlaid cutouts. In these cutouts, most gene sets have a score of or very close to 1 when calculated with the MM distance and pMM_0 distance score. However, with increasing values of α, the distances in this area change, diffusing from a dark, intensive red to a lighter, less saturated red indicating a decrease in distance.

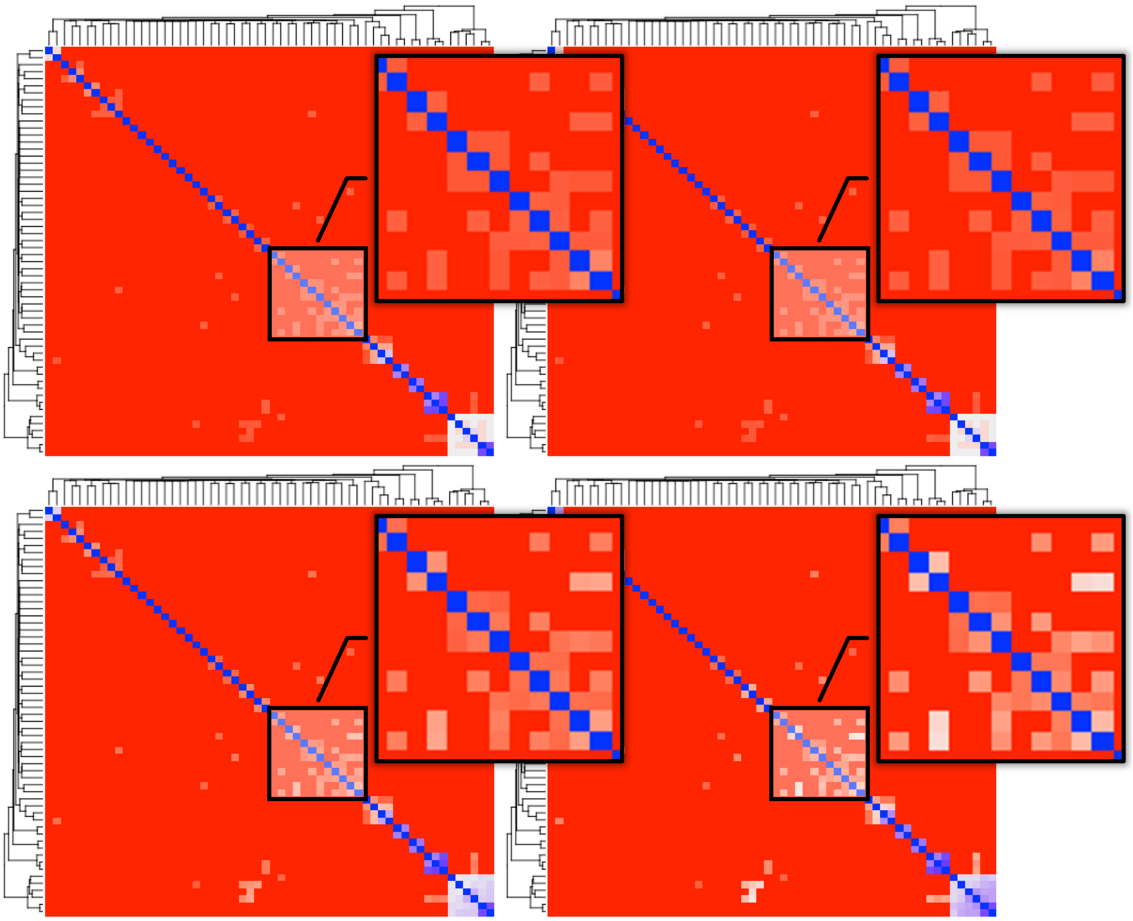


Figure 37 – Heatmaps of Different α Values
The figure shows the distance score comparison of different values of α, plotted as heatmaps. From left to right, top to bottom, the following is shown: the original Meet-Min distances of the gene sets, the pMM distances using an α of 0, 0.5 and 1. Additionally, the figure includes zoomed and highlighted cutouts of parts of the heatmaps, a feature than can not be directly achieved with `GeDi`'s functionality, but was rather used during figure assembly to enhance clarity and improve interpretation.

Overall, Figure 37 shows that the additional factor of protein-protein interaction in the pMM score can provide a more nuanced view of the biological functions underlying the data by capturing new and accentuating found similarities in the data. However, in the shown example, it was also observable that this effect might not be as strongly as one might expect, which was probably due to the additional filtering and overall reduced size of the selected gene sets for this comparison.

3.3.4.2 Aggregation of the Gene Sets using Clustering

Following the calculation of the distance scores, the data was clustered in the *Clustering Graph* panel. In the 'Select the Clustering Method' box (Figure 38A), I selected the clustering algorithms as well as individual clustering thresholds (Figure 38B) to evaluate the different clustering results. From the drop-down menu in the box, the distance scoring results to be used were chosen (Figure 38C). Once the clustering results were calculated, the available visualisations could be observed in the 'Geneset Cluster Graphs' box (Figure 38D).

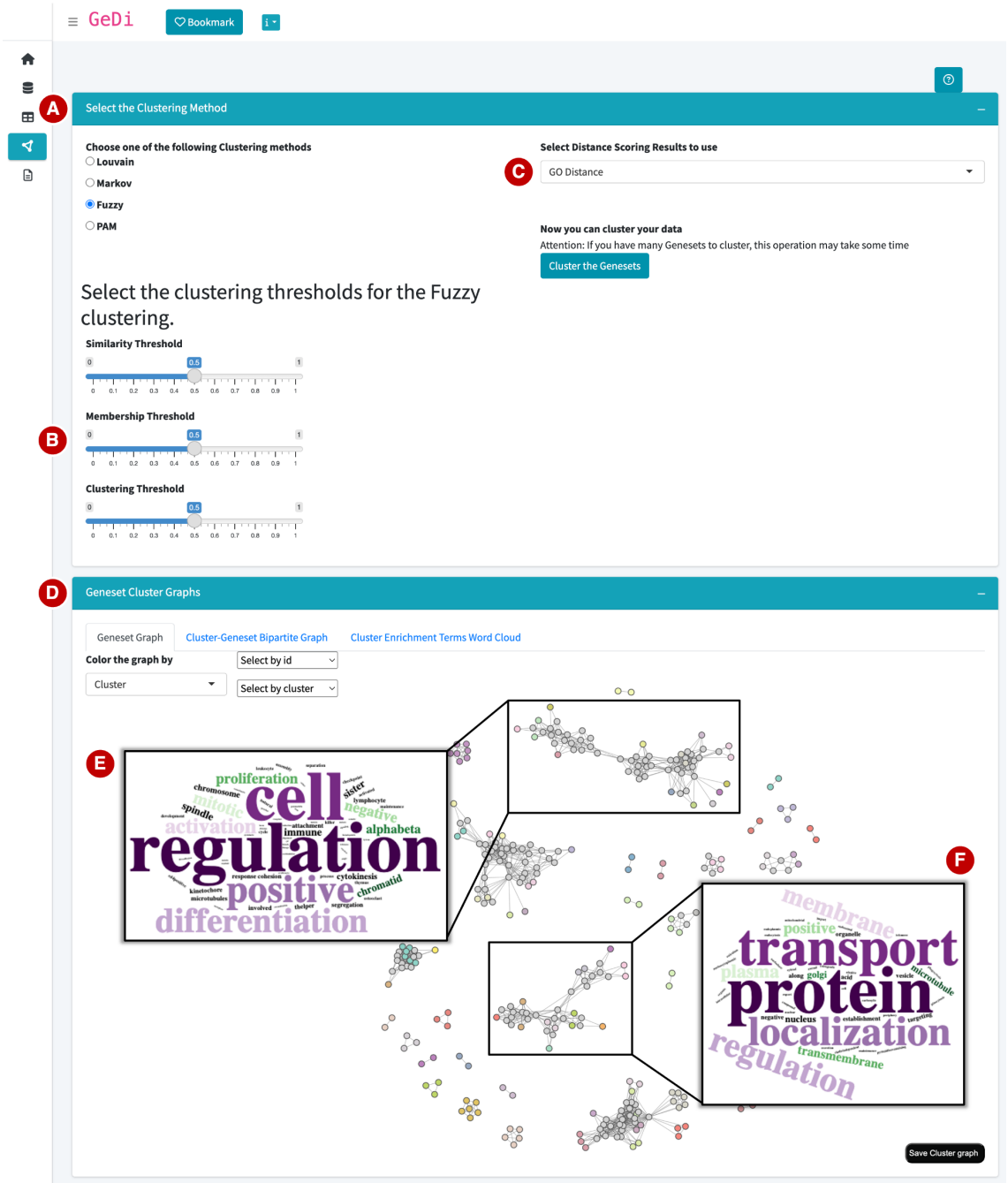


Figure 38 – The *Clustering Graph* Panel of GeDi
The figure shows the *Clustering Graph* panel of GeDi, with the 'Select the Clustering method' box (**A**), in which the clustering algorithm and thresholds (**B**) can be chosen and the clustering results be computed, based on the chosen distance scoring results (**C**). The computed clustering results are afterwards visualised in the 'Geneset Cluster Graphs' box (**D**). In **E** & **F**, word clouds of the highlighted clusters can be seen as zoomed overlays. While word clouds can be generated with GeDi, this combined and overlaid feature is not directly available within the app in this exact manner, but was rather arranged during figure assembly to enhance clarity and provide additional information.

After evaluation of the individual clustering results, I decided to highlight the results of the Louvain and Fuzzy clustering algorithm in this thesis to showcase the different properties of the individual clustering algorithms and the functionality of GeDi. For the Louvain algorithm, a similarity threshold of 0.5 was chosen, and for the Fuzzy clustering algorithm, I chose the value 0.5 for the similarity, membership and clustering threshold, as these values balanced the overall cluster composition.

In Figure 39, the clustering result of the pMM score using the Louvain clustering algorithm can be seen. In order to properly represent and distinguish the individual clusters, the colour was chosen to represent the cluster membership of each gene set. In the figure, it can be seen that the Louvain algorithm clustered the Treg dataset into several larger clusters and a variety of smaller clusters of two to three nodes/gene sets. A closer inspection of the larger clusters using the tooltip feature of the graph showed that the Louvain clustering algorithm divided the Treg dataset into 41 cluster, with overarching biological themes such as cell division, T cell differentiation and protein localisation and transport. However, it should be noted that in all visualisations of the *Clustering Graph* panel in `GeDi`, singletons (i.e., clusters only consisting of one gene set) are omitted to enhance the clarity of the visualisations.

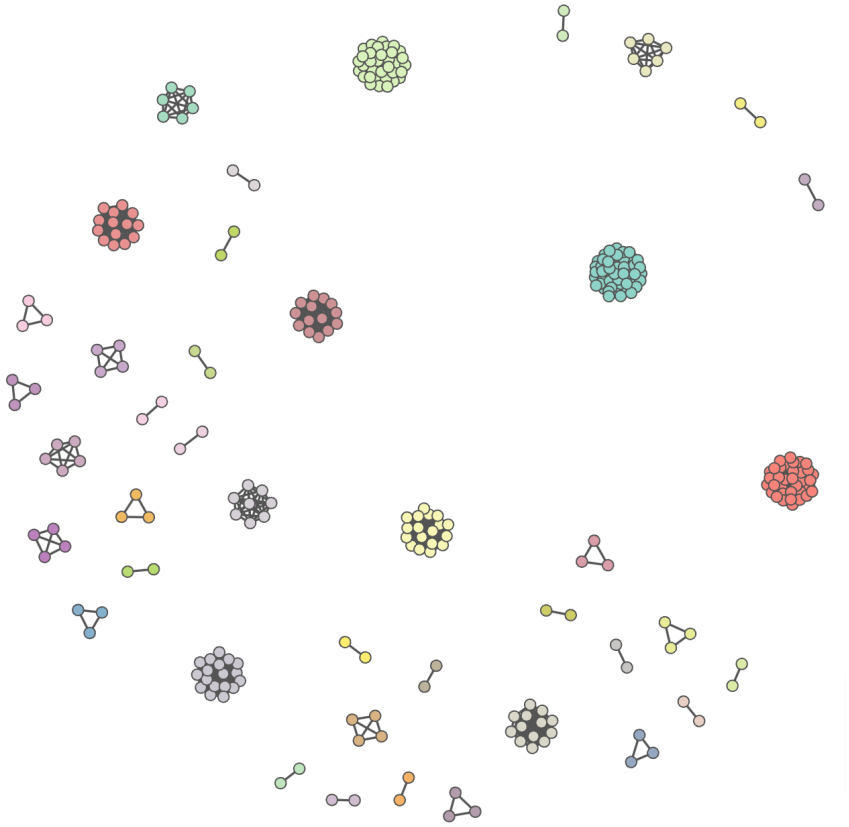


Figure 39 – Louvain Clustering of the pMM Distances
The figure shows the Louvain clustering results of the pMM distances. Each node represents a gene set, and the edges reflect the similarity between them. The node colour was chosen to represent the cluster membership of each gene set.

In a second investigation, the pMM distances were clustered using the Fuzzy clustering algorithm. The resulting clusters can be seen in Figure 40. The Fuzzy clustering algorithm divided the pMM results into 127 clusters, although a lot of these clusters shared at least some members. Due to this overlapping of clusters, the resulting graph visualisation gave the impression that the Fuzzy clustering algorithm resulted in several large, interconnected clusters and some smaller clusters of only a few gene sets. This impression was further supported by the colour of the nodes, which represented the cluster membership of each node. As the Fuzzy clustering algorithm allows nodes to be part of several clusters, a discrete colour palette, like the one used in `GeDi`, cannot fully reflect the cluster membership of each node, as multiple memberships cannot be

properly represented. Hence, the results should be interpreted cautiously when using the Fuzzy clustering algorithm and colouring the nodes by their cluster membership. In order to support the interpretation of the results, the hovering tooltip (see Figure 14 in Section 2.5.3) was used to explore the clusters in more details, specifically the largest, interconnected cluster in the middle of the shown network. This showed that the gene sets in these interconnected clusters are mainly involved in biological functions of cell division, especially involving the mitotic spindle, and T cell differentiation and proliferation; two larger biological themes, which had already been observed as clusters in Figure 39.

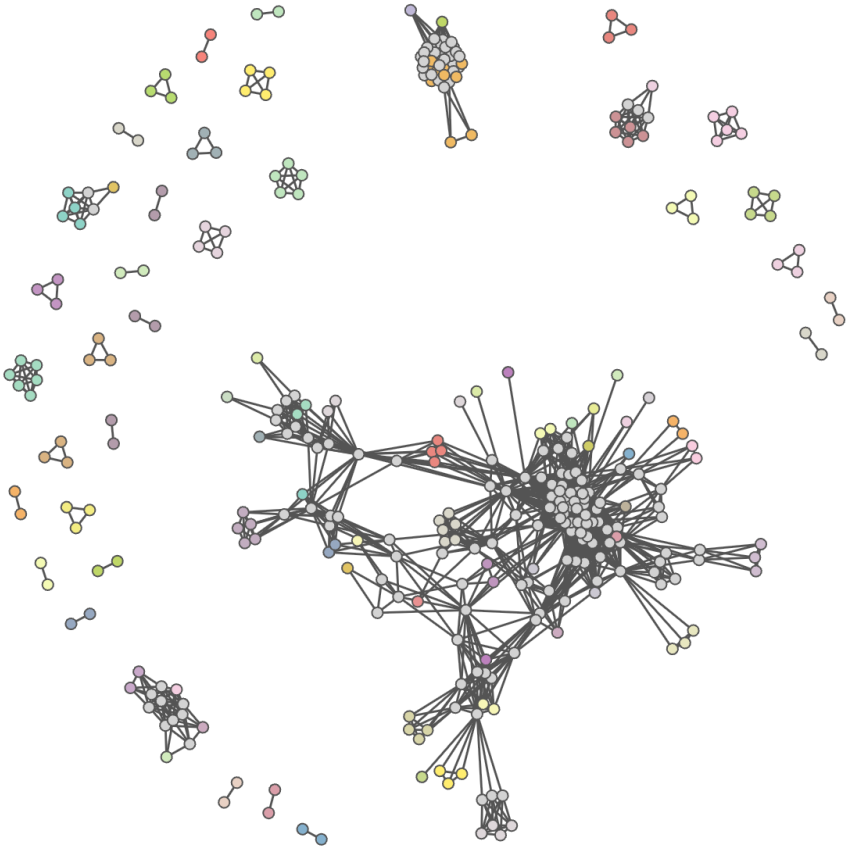


Figure 40 – Fuzzy Clustering of the pMM Distances
The figure shows the Fuzzy clustering results of the pMM distances. Each node represents a gene set, and the edges reflect the similarity between them. The node colour was chosen to represent the cluster membership of each gene set.

Following the results using the pMM distance score, the GO distances were used for clustering, in order to evaluate the different characteristics and behaviours of the clustering algorithms as well as distance metrics. In Figure 41, the clustering results of the Louvain clustering algorithm based on the GO distances can be seen. At first sight, the clustering results seem to be rather similar compared to the results shown in Figure 39. This observation was also reflected in the number of resulting clusters, with 49 final clusters for the GO-Louvain combination and 41 clusters in the pMM-Louvain combination.

In the heatmaps shown in Figure 36, it could already be observed that the pMM and GO distances were rather similar for the used dataset, which was now further supported by similar clustering results. Also, in terms of biological functions and pathways, the results were overall similar, with larger topics being cell division, T cell proliferation and protein localisation and transport.

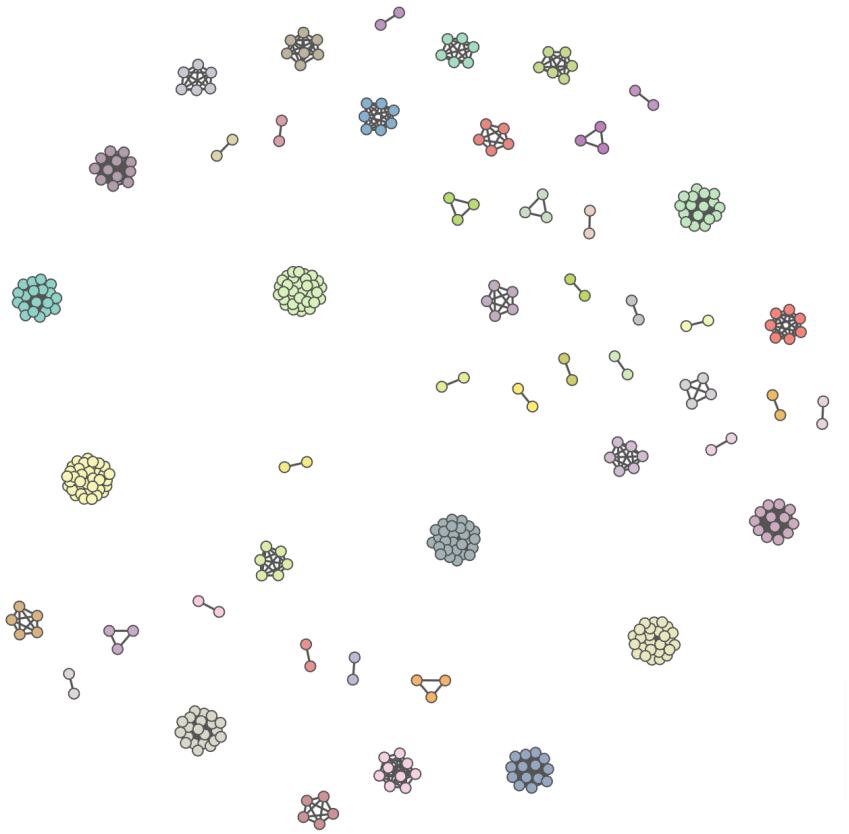


Figure 41 – Louvain Clustering of the GO Distances
The figure shows the Louvain clustering results of the GO distances. Each node represents a gene set, and the edges reflect the similarity between them. The node colour was chosen to represent the cluster membership of each gene set.

Finally, the GO distances were clustered using the Fuzzy clustering algorithm. The resulting clusters can be seen in Figure 38D. For these clustering results as well, a high similarity to the clustering results based on the pMM distances could be observed, with the GO-Fuzzy combination resulting in the identical number of clusters as the pMM-Fuzzy combination. Once again, it was observable that the clustering algorithm returned a few, strongly interconnected clusters and a variety of smaller clusters of only a few gene sets. Two of these large, interconnected clusters were further analysed using the other visualisations available in `GeDi`. In Figure 38E and F, word clouds of the highlighted clusters are shown.

As `GeDi` does not (currently) support the generation of a word cloud based on several clusters directly in the `Shiny` application, I first extracted the gene sets involved in these clusters using the tooltip feature of the graph representation. Afterwards, I used these lists of gene sets to generate subsets of the original functional enrichment results outside of `GeDi`'s `Shiny` application and plot word clouds based

on these subsets. The code for these word clouds, as well as the other figures shown in this thesis, can be found in the GitHub repository accompanying this thesis (`https://github.com/AnnekathrinSilvia/GSE130842_Showcase`) to ensure the reproducibility of the shown results.

The word clouds of Figure 38E and F show that the overarching themes of these clusters seem to follow the lines of cell differentiation and regulation, especially for alpha-beta T and immune cells (Figure 38E), as well as pathways of protein localisation and transport. These patterns were repeatedly found in the four different combinations of clustering algorithms and distance scores, highlighting their importance in the studied dataset.

The comparison of clustering algorithms revealed that, despite variations in cluster numbers and composition, consistent biological themes were be found in the data. Interestingly, it seemed like the results were more influenced by the choice of clustering algorithm than by the underlying distances, suggesting that the pMM and GO distances were quite similar for the Treg dataset.

3.3.4.3 Reproducibility using the Report Feature

A usual data exploration session using `GeDi` is intended to end with the generation of an HTML report, summarising the different exploration steps and results of the dataset. This report can be further enhanced using the bookmarking feature, with which individual gene sets and clusters can be bookmarked in the *Distance Scores* and *Clustering Graph* panel.

During the exploration session on the Treg dataset, I bookmarked various gene sets and clusters using the 'Bookmark' button, which can be found in the header of the application (Figure 42A). The bookmarked entities are summarised in the *Report* panel. This panel features intuitive visual summaries of the amount of bookmarked gene sets and clusters (Figure 42B & C) as well as tables listing the bookmarked entities. The tables provide additional information for the bookmarked gene sets and clusters, such as the description of the gene sets (if available in the input data), the member gene sets and their descriptions for the individual clusters.

In the presented use case, gene sets associated with various T cell pathways, such as "alpha-beta T cell differentiation involved in immune response", "regulation of T cell differentiation" and "CD4-positive, alpha-beta T cell activation", were bookmarked. Additionally, I bookmarked the largest cluster from the previously highlighted, interconnected clusters in Figure 38D, which were clusters 20 and 35. The gene sets in these clusters were mainly involved in pathways of T cell differentiation and protein localisation. In a last step during the exploration of the data, the *Report* panel was used to generate an HTML report via the 'Start the generation of the report' button (Figure 42D).

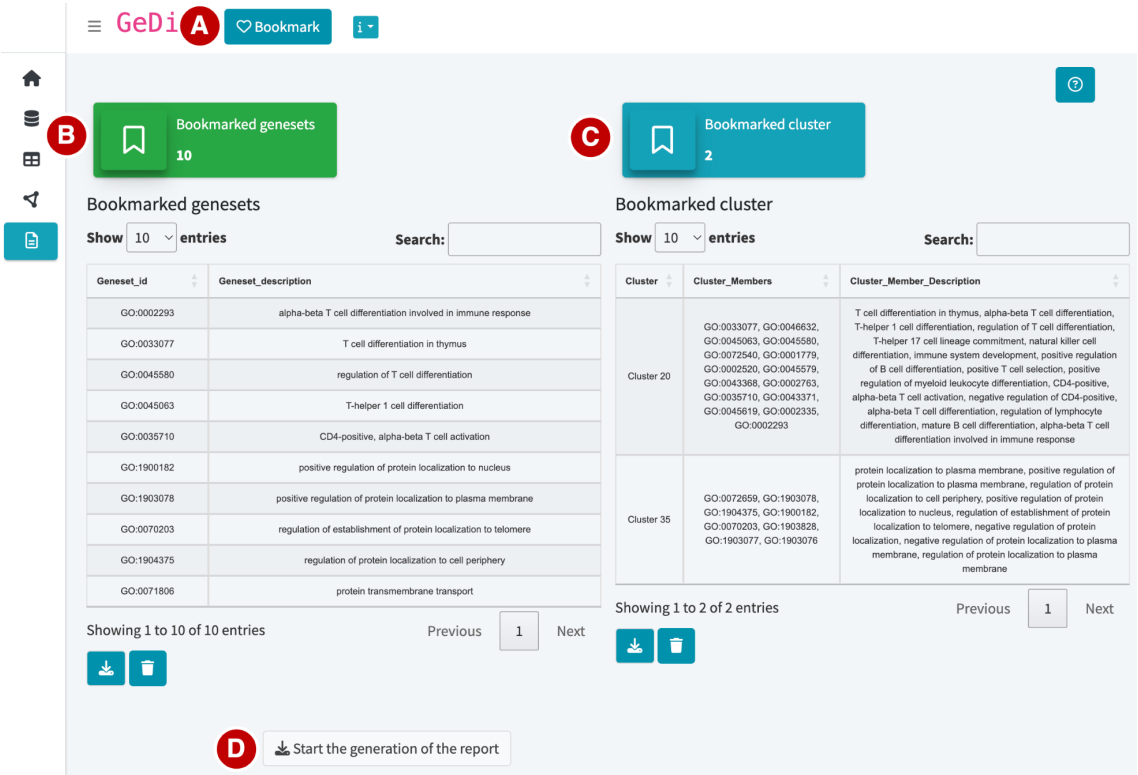


Figure 42 – The *Report* Panel of GeDi
The figure shows the *Report* panel of GeDi, which summaries the entities bookmarked in other panels of the application using the 'Bookmark' button (**A**). Small boxes provide a brief summary on the number of bookmarked gene sets (**B**) and clusters (**C**). This panel also supports the generation of an HTML report via the 'Start the generation of the report' button (**D**).

As most research endeavours are usually not the work of a single person but involve collaboration within and across various research groups, an easy way to share results and findings is desired and needed. In GeDi, this was addressed by the report feature. The generated HTML report features sections for each of GeDi's panels, highlighting the key results and exploration steps, as well as a dedicated section on the bookmarked gene sets and clusters. The report concludes with a section documenting the session information, containing information about the R environment, such as the used R version and all package versions of the local machine. The GitHub repository accompanying this thesis (`https://github.com/AnnekathrinSilvia/GSE130842_Showcase`) includes not only a script documenting all of the data processing steps, but also the HTML report of the described GeDi session to ensure full reproducibility of the presented results. Additionally, this is intended as a means to highlight the convenience and importance of the application's report feature.

With the functionality detailed in this thesis, GeDi aims to enhance the reporting standards for functional enrichment analyses in published research, while also advancing the interpretation of these results to drive new research directions and hypotheses. The exploration of the Treg dataset demonstrated how GeDi efficiently clusters and interprets functional enrichment results, consolidating the initial 500 gene sets into fewer, meaningful clusters.

This aggregation revealed overarching biological themes, such as T cell differentiation as well as protein localisation and transport. While these might have already been implied from top hits of the functional enrichment results list, such as "cell division" and "intracellular protein transport", these themes have been refined through the aggregation in GeDi. This provided an overall better overview of the data, as even smaller, more specific pathways were highlighted in the results. These might otherwise have been overlooked or lost in the extensive list of functional enrichment results. Consequently, the exploration of the dataset with GeDi provided a more comprehensive understanding of the underlying biological pathways and patterns.

4 Discussion

In this thesis, I presented my R package `GeDi`, demonstrating how it can be used to streamline the exploration and interpretation of functional enrichment results. Through a comprehensive literature review, I confirmed a concerning lack of proper documentation and reporting of functional enrichment analysis procedures across published research, which highlighted and validated the need for tools like `GeDi`. Additionally, this thesis explored the significance of standardised analysis workflows and the benefits such workflows offer, including enhanced transparency, reproducibility, and efficiency in data analysis.

In this chapter, I will summarise and reflect on the key aspects of the work presented in this thesis. Section 4.1 will emphasise the importance of standardised workflows and discuss how we addressed these in our published manuscript [Ludt et al., 2022]. On the note of standardised procedures, Section 4.2 will revisit the conducted literature review, broadening its implications and discussing future work planned in this context. Afterwards, Section 4.3 will focus on the design choices of the `GeDi` package and compare the work to similar tools, before Section 4.4 will discuss the results found in the showcase. Lastly, in Section 4.5, I will touch upon the current limitations of the package, proposing solutions and future enhancements of the work.

4.1 The Importance of Standardised RNA-Sequencing Analysis Workflows

In recent years, advancements in high-throughput sequencing have made these technologies increasingly accessible and feasible. As a result, sequencing data has become an integral part of research across many fields, with numerous published studies incorporating different types of sequencing data [Van Den Berge et al., 2019; Delacher et al., 2020; Huang et al., 2023; Kodali et al., 2023]. Consequently, the volume of data generated has grown rapidly, posing the need for efficient methods for processing, analysing and interpreting these datasets [Girolami et al., 2006; Akhmedov et al., 2020]. In order to manage this complexity and the amount of data, it is essential to adopt standardised workflows, irrespective of the study or research question at hand. Such workflows not only ensure the reproducibility of analyses but also promote consistency across different studies and datasets.

In this thesis, I have presented a standardised analysis workflow for bulk RNA-sequencing data, as detailed in our publication [Ludt et al., 2022]. The workflow simplifies the complex process of analysing bulk RNA-seq data by offering clear, well-documented, step-by-step analysis protocols. The article includes three *Basic Protocols* - exploratory data analysis, differential gene expression analysis and interpretation of the results - following the usual data analysis steps of bulk RNA-seq data (Figure 2 & Figure 15). By structuring the article and workflow in this way, we were able to provide detailed documentation for each step without burdening the reader or sacrificing readability.

This division of the workflow is also reflected in the three R packages discussed in the article - `pcaExplorer` [Marini, Binder, 2019], `ideal` [Marini et al., 2020] and `GeneTonic` [Marini et al., 2021]. Each of the *Basic Protocols* of the manuscript introduces one of

the packages and its included, interactive `Shiny` application. These applications enable users to interact with and explore RNA-seq data without needing to write R code themselves.

By offering a user-friendly interface, even those with minimal coding experience can perform complex analyses effortlessly. This transition from manual coding to interactive applications not only simplifies the workflow but also makes it more accessible to a broader audience. Additionally, these interactive tools foster collaboration by providing an easy-to-use platform that multiple researchers can engage with, ensuring that analyses are performed consistently and reproducibly across different teams [Supek et al., 2011; Akhmedov et al., 2020; The Galaxy Community, 2024].

In order to further support collaboration and reproducibility, each of these `Shiny` applications features an option to generate HTML reports. These reports document the steps taken in the analysis workflows, including code snippets and details on the software packages used. These reports not only guarantee the reproducibility of the analysis but also facilitate collaboration by providing a shareable format that can be easily reviewed by others.

Additionally, to foster and enhance the transparency of the analyses, the reports include code chunks of the underlying R code, as well as a summary of all used R software packages and versions at the end of each report, following current best practices for reproducible research [Stodden et al., 2013b; Markowetz, 2015; Munafò et al., 2017]. This approach aids users with limited coding experience by allowing them to see the code in context, fostering a learning experience. For more experienced users, the code snippets serve as a foundation that can be customised or extended for specific research projects, thus combining the convenience and ease-of-use of `Shiny` apps with the flexibility and adaptability of direct coding.

With `GeDi`, I developed an additional package that seamlessly integrates into the existing workflow presented in Ludt et al. [2022]. In order to achieve this integration, `GeDi` also accepts `GeneTonicList` objects as possible input format, effectively offering a potential "*Basic Protocol 4*" dedicated to the exploration and interpretation of functional enrichment analyses (Figure 20). This addition addresses a former gap in our protocols, where users previously still resorted to manual inspection of the functional enrichment results.

Besides the extension of our presented standardised workflow, we are continuously exploring ways to extend the functionality of the existing packages to align with recent research advancements. While bulk RNA-sequencing still remains a staple across a wide array of research scenarios, its single-cell counterpart has gained increasing attention in recent years. Considering the advancements seen in bulk RNA-seq, it is likely that similar, future improvements will also tackle current scRNA-seq challenges, such as the problem of generally high costs and data sparsity, potentially making it the new standard in RNA-sequencing within a few years [Lähnemann et al., 2020; Boakye Serebour et al., 2024]. To keep pace with these developments, our research group plans to extend the functionality of all packages in the workflow to also accommodate scRNA-seq data.

Furthermore, we continually update our workflows by integrating new methods and refining the existing code base, for example, through the recent introduction of our R/Bioconductor package `mosdef` [Dammer, Marini, 2024] into the code base. `mosdef` provides functionality for the most widely used steps in differential gene expression analysis workflows - such as the calculation of differentially expressed genes and functional enrichment results or the generation of various visualisations - thus providing a unified interface for these common steps. As such, the `mosdef` package consolidated and replaced formerly duplicated code across the three packages `pcaExplorer`, `ideal` and `GeneTonic`. This simplified the overall code structure and lead to easier maintenance of the whole code base, as changes and updates now only need to be applied once, thereby promoting consistency and reducing redundancy.

Through these ongoing efforts, we aim to maintain and refine our standardised workflow, ensuring not only reproducibility and transparency but also adaptability to recent research developments.

4.2 Assessing the Reporting Standards of Functional Enrichment Analyses

The literature review conducted in this thesis confirmed the gap for an easy-to-use, widely available tool for the exploration and interpretation of functional enrichment analysis results and validated the need for tools like `GeDi`.

Out of the 56 articles including functional enrichment analysis, less than 25% sufficiently documented their methods, indicating that most results could not be reproduced based on the provided information. In this context, there was a slightly higher standard observed in articles published in the high-impact journals *Cell*, *Nature* or *Science*. But still, more than half of the reviewed articles from these journals failed to specify critical details such as the statistical tests or software functions used, hence impeding reproducibility of the results.

All of the articles from these journals, however, included at least broad levels of documentation, whereas in the earlier literature searches, some articles lacked details entirely. Additionally, evaluating the introduced completeness score revealed that both articles fulfilling all nine of the evaluated criteria were published in these journals, as well as two out of three articles that had a completeness score of 8 points. But this was only a small fraction of all the reviewed articles, as 91% fulfilled at most 7 of the evaluated criteria, and 45% had 5 points or less.

Certainly, not every evaluated criteria is equally important for the reproducibility of the results — for instance, a detailed description of the method used is more critical than a visual representation of the results. While this might not be properly reflected by the simple scoring scheme applied, it still showed that the majority of articles missed critical information in their documentation, when ideally they should fulfill all of the evaluated criteria.

Additionally, there was a clear trend in the approach used to select the subset of gene sets the authors highlighted in their article. Of the articles that provided a rationale for selection, over 60% used "top" or "p-value" as criteria, suggesting that only the most significant results were emphasised. This may indicate that researchers were overwhelmed by the amount of available results, which are usually returned as extensive, long lists of gene sets, ultimately leading to an incomplete analysis and the loss of valuable information and insights.

During this literature review, I also evaluated the code availability of the reviewed articles. Only 25% of the 56 articles provided access to the analysis code used. While a thorough documentation of the used methods, software and tests enables the replication of the presented results, having the analysis code available adds another important layer of reproducibility, potentially compensating for any missing information in the article itself. Furthermore, providing the analysis code promotes greater transparency and collaboration, enabling other researchers to validate findings and use the same methods in their own studies. It allows for the detection of potential errors, improves efficiency by eliminating the need to recreate analysis pipelines, and ensures consistency in results, ultimately supporting scientific integrity and progress [Stodden et al., 2013b; McKiernan et al., 2016; Goldacre et al., 2019; Page et al., 2022].

Overall, this literature review revealed that a consistent standard for reporting functional enrichment results is still lacking. It is evident that there is a clear need for tools that streamline the interpretation process. Researchers may be overwhelmed by the sheer volume of data produced by functional enrichment analyses, leading to an overemphasis on a narrow selection of results and potentially overlooking key insights.

In the future, our research group plans to extend this literature review to cover a wider range of publications, hence providing an even more thorough view on the current standard of functional enrichment reporting in published literature. The literature review in this thesis mainly focused on the documentation as well as the result presentation of the conducted analyses to evaluate the benefits that tools like `GeDi` could bring to the exploration and interpretation of functional enrichment results. In future work, we also plan to focus on the parameters evaluated in the work by Wijesooriya et al. [2022]: In contrast to the literature review presented here, Wijesooriya et al. focused their evaluation more on the correct execution of functional enrichment analyses, assessing factors such as background gene lists and multiple testing corrections. Combining both sets of evaluation parameters would allow for a more thorough assessment of both execution and reporting standards.

We also aim to use machine learning and large language models to improve the search for relevant literature. Using the search methods employed in this thesis, more than 40% of the sampled and reviewed articles did not include enrichment analysis results along the lines of the definition in Section 1.3, despite using queries of the type "Enrichment AND Analysis". A properly trained large language model could not only improve the search for articles, but also filter the ones not fitting the search criteria early on, hence leading to a more efficient selection of articles.

Furthermore, our group is currently working on an R/Bioconductor package aimed at facilitating the documentation process for functional enrichment analyses. While `GeDi` supports the exploration and interpretation of functional enrichment analysis results, it does not include functionality to perform the actual analysis nor to facilitate the documentation of the conducted analysis. In contrast, the new package is designed to ensure reproducibility by capturing all relevant metadata, such as software versions, background genes, and genome indices versions, directly within the result object during the functional enrichment analysis. Its core concept involves providing wrapper functions around currently available functional enrichment implementations in Bioconductor and CRAN (e.g., `topGO` or `clusterProfiler`), which execute these functions while also capturing the metadata. The package will also include functionality that allows the generation of "Material and Methods"-like summaries of the conducted analysis. These summaries could be directly included in publications, hence streamlining the analysis and documentation of functional enrichment analysis. Once finalised, this package will be seamlessly included in our standardised workflow. It will also be compatible with `GeDi`, hence ensuring that we provide users with a streamlined workflow to execute, document, explore and interpret functional enrichment analyses.

4.3 The Power of `GeDi`

In many research fields, a recurring question centers around the identification of pathways or biological functions affected under various conditions. This could involve the study of diseased versus healthy subjects, knockout versus control samples, or other multi-group experimental designs. To detect these pathways and biological functions, functional enrichment analysis is often the method of choice. With the rapid growth and advancement of high-throughput sequencing technologies, however, the amount of data needing to be processed, analysed and interpreted has skyrocketed.

An additional factor increasing the number and amount of results to analyse is the fact that there is usually not a one-to-one relationship between datasets and functional enrichment results, as the number of comparisons increases when multiple conditions are analysed. For example, while two conditions result in a single comparison, three conditions lead to three comparisons, and four require six, causing a quadratic increase in the number of comparisons as the number of conditions grows.

Interpreting these results to draw meaningful conclusions is often the most challenging step of omics data analysis, as it requires both, expertise and a considerable amount of invested time. However, while the demand for computational biologists and bioinformaticians rises, not every research group has the resources to hire dedicated specialists. Consequently, existing group members - who may lack sufficient training in bioinformatics - often find themselves responsible for data analysis. These situations can lead to a significant bottleneck in the data analysis process as researchers try to balance acquiring new technical skills with their ongoing work - an issue many collaborators in our network reported, highlighting the need for tools that facilitate the analysis and interpretation of complex datasets, particularly for those research group members who only possess minimal coding experience.

In order to address this problem and extend our suite of interactive R/Bioconductor packages for bulk RNA-seq analysis, I developed `GeDi`, a package designed to streamline and facilitate the interpretation of functional enrichment results. It offers both stand-alone functionality as well as interactivity via a `Shiny` application, making it accessible to a broad range of users, from those with minimal coding experience to proficient R users. Additionally, a demo version of the `Shiny` app is available online (`http://shiny.imbei.uni-mainz.de:3838/GeDi`), allowing users to explore the tool without needing to install R or the package itself. Although some features, such as the download of intermediate results or the generation of reports, are limited in the demo version, it still offers a robust solution for users who may lack the computational infrastructure to install the package locally.

`GeDi` is implemented as an R package and available on Bioconductor. This design decision was made early in the tool's development, driven primarily by the widespread use of R for the analysis of biological data, as reflected in the large number of available (Bioconductor) packages. Another important factor influencing this choice was the pre-existing suite of R/Bioconductor packages developed within our research group. `GeDi` emerged as part of our ongoing work, including our manuscript "Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic" [Ludt et al., 2022]. From the outset, a key requirement for `GeDi` was seamless integration into our workflow, in order to not only streamline and facilitate the interpretation of functional enrichment results, but also to extend our comprehensive suite of packages for the analysis of bulk RNA-seq data through an additional, and certainly needed, component. In order to achieve this, one of the available input data formats of `GeDi` is the `GeneTonicList` object; an input format which is already used in `GeneTonic`.

Furthermore, `GeDi`'s implementation as a Bioconductor package offers advantages beyond just seamless integration into our workflow. As part of its implementation as a robust R package, `GeDi` includes an extensive set of unit tests designed to verify its functionality, even in edge cases. At the time of writing, the package achieves 95% code coverage, meaning that most of the source code is executed and validated during testing. This test suite, which is executed regularly during Bioconductor's daily and weekly builds, ensures that both the `Shiny` application and stand-alone features function as intended and safeguards against unintended alterations of existing functionality during future updates and extensions.

The package's submission to Bioconductor also means adherence to rigorous package development guidelines, including structured and well-documented code, unit testing and a vignette detailing its functionality. Moreover, Bioconductor implements a peer-review process for new submissions, ensuring that each new package is carefully inspected by experienced developers, guaranteeing that it meets the community standards before inclusion. Once part of the community, packages benefit from the platform's biannual release cycle, which ensures the compatibility of the packages with the latest R versions as well as with each other. During the release cycle, packages are intensively checked for compatibility issues, and package developers are informed about necessary updates to their packages, ensuring that a consistent and stable Bioconductor environment is maintained. Additionally, being part of Bioconductor's ecosystem enhances `GeDi`'s inter-

operability with existing packages, broadening its potential user base and making it an accessible tool for a wide range of researchers.

The core concept of GeDi is centered on the calculation of distance scores for the gene sets of functional enrichment results, followed by clustering the data based on these scores. This approach was chosen as it is easy to understand and interpret, yet powerful enough to uncover underlying patterns and themes from the data. Additionally, GeDi incorporates biological network data from protein-protein interaction networks to account for the fact that biological processes and connections are not only observable at the gene level, but mostly take place at the protein level.

As distance metrics, six different available distance scores were chosen: the Meet-Min distance, the Jaccard distance, the Kappa distance, the Protein-Protein weighted Meet-Min distance, the Sørensen-Dice distance and the GO semantic distance score. The first three as well as the fifth score are solely set-based, usually quantifying the overlap between two gene sets and afterwards scaling this factor (e.g., by the combined size of both sets or the size of the smaller set). As such, these scores are easily calculated, interpreted and especially useful in scenarios where focus is on the gene overlap between gene sets. This could be studies in which the underlying difference between conditions is based on individual genes such as cystic fibrosis, sickle cell anemia or Huntington's Disease [Dayalu, Albin, 2015; National Heart, Lung, and Blood Institute, 2024; Xu et al., 2024]. The pMM score of Yoon et al. [2019] on the other hand, as an extension of the MM score, is based on a set component as well as a network component that includes protein-protein interaction information. Hence, this score can be used to capture the nuances of biological interactions and functions, which might be helpful in research endeavours where it is not clear from the start what is to be expected from the data. Lastly, GeDi implements the GO semantic distance, a distance score specifically based on and usable only for GO identifiers. This score can be applied to leverage the inherent interaction information of the GO database and its identifiers. As GO terms were the most commonly used identifiers found in the literature review (Section 3.2, Figure 16 to 18), it shows that the database is widely used in the community and hence the score can be valuable for a lot of research endeavours.

Additionally, GeDi integrates several clustering algorithms that can be broadly categorised by their clustering characteristics: those that partition data into distinct clusters (Louvain and Markov), those that allow for overlapping clusters (Fuzzy), and those that require a predetermined number of clusters (PAM). With these properties, the GeDi package can be applied to various kinds of biological data. The Louvain algorithm is particularly well-suited for large-scale networks, offering scalability and hierarchical clustering. Due to this, it is ideal for detecting large, cohesive gene or protein modules, such as those involved in immune responses or regulatory networks related to diseases like Alzheimer's [Lee et al., 2024; Yi et al., 2024]. The Markov clustering algorithm on the other hand is optimised for complex, interconnected biological networks. It efficiently identifies clusters by simulating random walks through a network, allowing densely connected areas to form clusters while sparse regions disappear. This makes it well-suited for data where clusters overlap or are noisy, as it can naturally separate them through its iterative process of expansion and inflation. Examples of applicable scenarios are the analysis of developmental stages, where gene or protein expression changes gradually

over time, but distinct stages (early vs. late) must be identified [Jones et al., 2020]. Another great use case is the analysis of protein-protein interaction network data from which larger protein complexes or pathways should be identified [Huttlin et al., 2017]. Fuzzy clustering on the other hand excels in cases where genes or proteins belong to multiple processes or functions, such as cancer biology [Zeng et al., 2021; He et al., 2023]. Lastly, PAM clustering is robust to outliers, works well for small to medium datasets, and is flexible in handling non-Euclidean distances, offering interpretability by using actual data points as medoids. It is especially valuable when clear, non-overlapping clusters are expected.

Given the increasing need for effective tools to interpret functional enrichment results, `GeDi` contributes to a broader array of tools and implementations designed to address this challenge. Many of these existing tools emphasise the visualisation of enrichment data, enhancing interpretability and facilitating deeper insights.

One well-known example is Enrichr, a widely used web-based platform that retrieves biologically enriched pathways based on input gene lists. It integrates databases such as GO and KEGG to provide a quick, user-friendly solution for functional enrichment analysis [Chen et al., 2013; Kuleshov et al., 2016]. Enrichr offers wide accessibility and user-friendliness, allowing real-time computations and result visualisation through an intuitive web interface. However, it differs from `GeDi` in several key aspects. For instance, Enrichr computes the functional enrichment itself, whereas `GeDi` primarily focuses on exploring and interpreting precomputed enrichment results.

Additionally, Enrichr requires users to input Entrez gene identifiers, limiting flexibility for other types of gene annotations, while `GeDi` allows a broader range of identifiers. While there is also the `enrichR` R package, which offers an interface to and the integration of Enrichr into scripted workflows, the tool focuses on the calculation of functional enrichment results. In contrast, `GeDi` is focuses on the interpretation and exploration of the results. Therefore, both tools should be seen as complementary, with each bringing its own unique strengths, and their combined use providing a more thorough and comprehensive way of analysis.

Another great example for an excellent tool is `GScluster`, which was developed alongside the pMM distance score by Yoon et al. [2019]. Like `GeDi`, `GScluster` is highly focused on the interpretation and reduction of redundancy in functional enrichment results. Both tools aim to group gene sets into clusters that represent distinct biological processes, facilitating easier exploration and interpretation of large datasets.

The work by Yoon et al. [2019] also features a `Shiny` application for interactive exploration of the results. However, their work has some limitations and drawbacks. While `GScluster` also offers the integration of PPI information into the analysis, their tool is limited in the number of species that can be used for the analysis. In fact, `GScluster` only supports the analysis of ten species, namely, human, mouse, rat, arabidopsis thaliana, C.elegans, E.coli, fly, rice, yeast and zebrafish, and does not provide the user with the possibility to download the respective PPI data, which limits its applicability to non-model organisms.

Additionally, GScluster only implements a limited set of distance metrics (MM, Kappa and pMM) and a single clustering algorithm (Fuzzy clustering from the DAVID database). This restricts its applicability, as these options may not be suitable for all data types and analyses. GeDi on the other hand, implements six different distance metrics and four different clustering algorithms to broaden its area of applicability.

With its functionality, GeDi aims to address a key bottleneck in the exploration and interpretation of functional enrichment analyses by offering tools that facilitate a more streamlined and efficient analysis process. Unlike existing solutions, GeDi provides a broader range of features that allow users to aggregate extensive lists of functional enrichment results into clusters of functionally related gene sets.

This aggregation simplifies the interpretation by highlighting overarching biological themes, reducing redundancy, and making it easier for researchers to identify key patterns and insights within complex datasets. By doing so, GeDi not only enhances the clarity of results but also accelerates hypothesis generation and further exploration.

4.4 Exploring GeDi's Impact on Transcriptome Data Interpretation

This thesis demonstrated the functionality of GeDi through the analysis of a publicly available murine bulk RNA-sequencing dataset. The data, originally published by Delacher et al. [2020], consisted of 56 bulk RNA-seq samples of various tissues and conditions. After thorough discussions with the original authors, I focused on comparing the Klrg1-negative Nfil3(GFP)-negative Tregs ($\mathrm{Klrg1^{-}Nfil3^{-}}$) and Klrg1-positive Nfil3(GFP)-positive Tregs ($\mathrm{Klrg1^{+}Nfil3^{+}}$) samples to demonstrate the advantages of standardised workflows and showcase the benefits and functionality of GeDi.

Initial exploratory data analysis using pcaExplorer confirmed the high quality of the data, showing consistent read counts across samples (Figure 22D). The PCA plot (Figure 23A) further indicated that the samples' condition was the primary source of variation, aligning with the original experimental design described by Delacher et al. [2020]. During this analysis, the expression levels of Areg, Il10, Klrg1, and Nfil3 were examined across the two conditions, showing higher expression in $\mathrm{Klrg1^{+}Nfil3^{+}}$ compared to $\mathrm{Klrg1^{-}Nfil3^{-}}$. These findings align with previous work by Delacher et al. [Delacher et al., 2017, 2020], which identified Areg and Il10 as key marker genes of the tisTregST2 subpopulation. Both $\mathrm{Klrg1^{+}Nfil3^{+}}$ and $\mathrm{Klrg1^{-}Nfil3^{-}}$ are precursor cells of this subpopulation, with $\mathrm{Klrg1^{+}Nfil3^{+}}$ representing a later precursor stage, thus displaying gene expression patterns more similar to the tisTregST2 cells.

Subsequently, the differentially expressed genes as well as enriched pathways were determined using ideal. The analysis identified roughly a quarter of all expressed genes as differentially expressed, with the top hits being Ctla4 and Ccr7, which are genes involved in negative regulation of T cell response and positive regulation of immune response, respectively. Klrg1 was identified as the gene with the highest positive log2FC, again showing consistency with the experimental design of the data, where its expression was used to distinguish Treg cell subpopulations [Delacher et al., 2020]. Other genes

with high, positive log2FC included Cc2d2a, Ccr10 and Tigit. These genes are involved in processes of protein localisation to ciliary transition zone, chemotaxis and negative regulation of T cell activation. They also highlight the differences in gene expression between the two subpopulations of Treg cells in the samples, showcasing the different expression profiles which arise due to the differentiation of the cells into the tisTregST2 subpopulation.

Furthermore, a functional enrichment analysis was conducted on the data using the `topGO` package from within `ideal` (see Figure 29). This analysis revealed "cell division" (GO:0051301) as the affected pathway with the smallest p-value. However, among the top hits were also "positive/negative regulation of apoptotic processes" (GO:0043065, GO:0043066), "positive regulation of T cell cytokine production" (GO:0002726) and "intracellular protein transport" (GO:0006886). These pathways suggest that the different subpopulation of Treg cells in the samples are undergoing differentiation towards the more specialised tisTregST2 subpopulation. Afterwards, the results were explored using `GeneTonic`, which provided first insights into the connection and relationships between the individual gene sets of the functional enrichment results.

In the next phase, the functional enrichment analysis results were analysed in-depth using `GeDi`. Initially, in the *Data Input* panel of `GeDi`, gene sets with a size of more than 200 genes were filtered to prevent hub gene sets from distorting the clustering analysis and to remove very broad and general terms, such as "biological_process", "cellular biosynthetic process" or "organelle organization". In order to highlight different distance metrics, I chose the pMM as well as the GO distance score to present in this thesis. The calculated distances showed an overall high similarity, with most gene set pairs having a distance close to or of 1 (see Figure 36F & G). Around the diagonal, there were differences observable between the two chosen distance scores with the GO distance score having larger areas of white and light-blue colour (i.e., distances around a value of 0.5). This indicated that these gene sets had smaller distances when scored with the GO distance score compared to the pMM score.

Clustering was performed using the Louvain and Fuzzy clustering algorithms. Once applied to the Treg dataset, their individual characteristics could be nicely explored, with the Louvain algorithm partitioning the data into a few larger, distinct groups, while the Fuzzy algorithm generated numerous highly interconnected clusters (see Figure 38 to 41). However, the differences in distances noted in the heatmaps were less apparent in the clustering results. In the shown results, the number and overall composition of clusters seemed to be more influenced by the choice of clustering algorithm compared to the underlying distance score used. This suggested that the chosen clustering thresholds of 0.5 were too small to capture the differences between the calculated distances. Hence, it could be interesting in the future to evaluate various different clustering thresholds to assess the influence of the chosen distance metric on the resulting aggregation of the data.

Across all presented combinations of distance metrics and clustering algorithms, common biological themes and patterns were be observed, with the most prominent related to differentiation and proliferation of T cells as well as protein localisation and transport. This aligned with the biology of the Treg dataset, as Delacher et al. [2020] identified both

$\mathrm{Klrg1^{-}Nfil3^{-}}$ and $\mathrm{Klrg1^{+}Nfil3^{+}}$ Treg cells as precursors of tissue-resident Tregs (also called tisTregsST2), a subpopulation involved in tissue regeneration [Delacher et al., 2020]. In this lineage, increasing expression of Klrg1 and Nfil3 initiated Treg cell differentiation from the earlier $\mathrm{Klrg1^{-}Nfil3^{-}}$ state to the later $\mathrm{Klrg1^{+}Nfil3^{+}}$ stage, hence explaining the large variety of pathways associated with the differentiation and proliferation of T cells. With ongoing differentiation, these subpopulations of T cells also change the type and amount of surface proteins and antigens they present, explaining the prominent topics of protein transport and localisation in the results of the analysis [Delacher et al., 2020].

Additionally, I evaluated the effect of different α values on the pMM score in this thesis. For this analysis, the data was filtered to include only gene sets smaller than 10. This filtering aimed to shift the emphasis away from the set-based component of the pMM score, which tends to be higher for larger gene sets with numerous associated genes, and instead highlight the PPI-based component. By focusing on smaller gene sets, the PPI interactions were expected to have a stronger influence on the score, given the limited potential overlap among genes.

The evaluation revealed only minimal differences across the various α levels (see Figure 37). However, an overall trend could be observed, where gene sets with initially smaller MM distances showed further reductions in distance when PPI information was incorporated (see the zoomed section of Figure 37). However, there were no gene set pairs observable in the heatmaps, whose initial distance score of 1 was reduced based on PPI information alone. This indicates that for the data used in this thesis, the protein-protein interactions observed are highly connected to the shared and overlapping genes in the gene sets.

Overall, the showcase of this thesis demonstrated how our published standardised analysis workflow can be used to streamline the analysis of published bulk RNA-sequencing data. Additionally, the thesis showed how `GeDi` can be seamlessly integrated into this workflow, improving the interpretation and exploration of functional enrichment analysis results. During the exploration, `GeDi`identified similar biological patterns of T cell differentiation and proliferation as the original authors, but also highlighted previously unexplored topics of protein localisation and apoptotic pathways. These findings showcase how `GeDi` can facilitate the exploration of functional enrichment results and provide new directions for future research endeavours.

4.5 Limitations and Outlook of `GeDi`

During the development of my package `GeDi`, several design choices were made, particularly the use of R and Shiny, and the integration into the Bioconductor ecosystem. This choice was not only influenced by the existing R/Bioconductor infrastructure in our group, aiming for `GeDi` to be seamlessly integrated into our standardised workflow, but also by the high prevalence and use of R for the analysis of biological data across a wide variety of research areas, provided by R's extensive framework and diverse package support. However, while R is commonly used for the analysis of biological data, certainly not all research groups use this framework, hence restricting the accessibility of `GeDi`. To improve accessibility, one consideration is hosting the package on Galaxy, which is

a widely used open-source platform for bioinformatic analyses [The Galaxy Community, 2024]. Galaxy's architecture supports the integration of R packages by converting the R scripts into Galaxy tools; a set of files informing Galaxy how to run a particular analysis. Developing and hosting a GeDi Galaxy tool could broaden the overall reach of GeDi, while still leveraging R/Bioconductor's benefits.

GeDi offers a variety of distance metrics and clustering algorithms. While this was an implementation choice I specifically took to reduce bias in the analysis and interpretation of the results, it should not be overlooked that this variety might inadvertently encourage selective interpretation or "cherry-picking" of results, thus leading to false-positive results. As demonstrated in Section 3.3.4, the different distance scores and clustering results can lead to different results, either in the calculated distances or the number and composition of the resulting clusters. In this thesis, I compared the pMM and GO distance score, as well as the results of the Louvain and Fuzzy clustering algorithms with a murine bulk RNA-seq dataset. When observing the heatmaps shown in Figure 36, it is apparent that both scores result in varying but overall similar heatmaps, which is also reflected by the clustering results (Figure 38 to 41), in which similar biological themes and patterns could be found. However, it is important to acknowledge that this consistency may not apply to all datasets, and significant variations can occur. While exploring different settings and parameters in GeDi (e.g., adjusting the value of α or the distance thresholds) is encouraged, it is essential to approach the data without preconceived expectations to avoid biased interpretations.

Users should also exercise caution when comparing results, especially the distance heatmaps available in GeDi. In the Shiny application of GeDi, the rows and columns of the heatmaps are by default clustered by their calculated distances, which can result in varying order and arrangements of the gene sets, making direct comparisons challenging. Although the stand-alone version of the **distanceHeatmap()** function allows users to disable clustering, this option is not yet available in the Shiny interface. To address this, future versions of the app will include a feature to disable clustering of rows and columns in the heatmaps. As an intermediate solution to ensure that the heatmaps shown in this thesis, specifically in Figure 36 and Figure 37, are comparable, the rows and columns of the resulting heatmaps were sorted to maintain a consistent gene set order, hence ensuring that the heatmaps were directly comparable.

In GeDi, reproducibility is ensured through the report generation feature. However, in the current implementation of the package, the report will only capture the latest state of the application, i.e., the selected distance metric and clustering algorithm from the *Distance Scores* and *Clustering Graph* panel. Hence, the report currently does not record all calculated distance scores or clustering results. While this may suffice in cases where users are content with their final choices, comprehensive reporting of all analyses would enhance reproducibility and reduce the risk of cherry-picking. Additionally, improvements could include dynamic tables in the report, like linking bookmarked gene sets and clusters to relevant databases for further exploration. Nonetheless, the current version of the report has received positive feedback in the user community and from our collaboration partners.

While GeDi benefits from its integration into the R and Bioconductor communities, leveraging widely adopted data structures and packages for easy incorporation into existing workflows, it also faces challenges related to package dependencies. As dynamic and evolving communities, package features and functions may be updated or deprecated over time, potentially affecting GeDi's functionality. There are several ways to address this challenge. One approach would be to reimplement certain features directly within GeDi, although this is impractical for larger, essential packages such as Shiny, igraph or the plotting framework of ggplot2. A more practical solution is the creation of isolated environments where package versions remain consistent, ensuring reliable reanalysis and stable functionality. This can be achieved using the renv package [Ushey, Wickham, 2024], which manages dependencies and locks specific package versions. The environment details are stored in a lockfile, which can be easily shared between collaborators, enabling the recreation of the same environment across different systems.

Additionally, future versions of GeDi aim to minimise package dependencies to reduce the risk of disruptions in functionality. In order to achieve this, core functionality might be directly implemented within the package to avoid reliance on external packages. Furthermore, existing dependencies will be reviewed for overlapping functionality, and efforts will be made to replace large dependencies with more lightweight, stable alternatives. Where feasible, functionality provided by external packages could be substituted with equivalent implementations using functions available in the base implementation of R to further streamline the package.

GeDi is specifically designed to handle large lists of functional enrichment results. However, these large lists of gene sets present their own challenges, particularly regarding memory usage and runtime. This is especially true for large datasets where calculating distances, especially with the pMM score, demands substantial computational resources. To address this, the **BPPARAM** argument was introduced in each distance scoring function enabling the parallel execution of calculations, thus significantly improving performance. However, parallel computing is currently not supported outside of the stand-alone version of the distance scoring functions and currently under rework, because certain combinations of operating system and parallel backend lead to instabilities in the Shiny application. The feature will be reintroduced to the Shiny app, as soon as stability is ensured for all commonly used operating systems. Additionally, further improvements in computational efficiency are planned for future versions of GeDi. These may include optimising the way the PPI data is handled, such as replacing the current `data.frame` object with a more efficient data structure that allows faster lookup times for its entries. By doing so, GeDi could handle larger datasets more efficiently, further enhancing its utility for analysing complex biological data.

An additional extension planned for GeDi involves expanding the available databases for the download of protein-protein interaction data. Currently, GeDi only supports the STRING database, which is commonly used and offers PPI data for a wide range of organisms. However, some users may prefer or require data from alternative PPI sources, such as BioGRID and IntAct [Oughtred et al., 2021; del Toro et al., 2022]. To address this, our group is developing a new R/Bioconductor package called NetworkHub, which will provide functionality for retrieving, caching, processing, and preparing PPI networks and their associated information from multiple databases. Once NetworkHub is fully developed and

available on Bioconductor, GeDi will integrate its functionality, offering users greater flexibility in accessing PPI data from various sources. Readers can find the current development of NetworkHub on GitHub (https://github.com/lottawagner/NetworkHub).

In conclusion, this thesis demonstrates the power of GeDi in streamlining the analysis of functional enrichment results, highlighting its utility in interpreting complex biological data. While showcasing the tool, interesting clusters of functionally related gene sets could be identified, which illustrated how GeDi can be used to uncover interesting patterns and connections in the data, which are not as easily recognisable in a manual interpretation of the functional enrichment results.

With the features explored in this thesis, GeDi will hopefully contribute to an overall improvement of the reporting standards of functional enrichment analyses in published research, as well as advance the interpretation of such data, leading to new research endeavours and hypotheses.

Zusammenfassung

In den letzten Jahren hat sich die Bulk-RNA-Sequenzierung (RNA-seq) als Goldstandard für die Transkriptomanalyse etabliert. Dadurch ist die Menge an generierten Daten und Ergebnissen erheblich gestiegen. Diese Entwicklung erschwert jedoch zunehmend die Interpretation, insbesondere bei der funktionellen Anreicherungsanalyse (hiernach Functional Enrichment Analyse genannt). Functional Enrichment Analysen sind ein grundlegender Schritt in der Analyse verschiedener Omics-Datensätze. Ihr Ziel ist es, differentiell regulierte Signalwege zwischen experimentellen Bedingungen aufzudecken und Einblicke in die zugrunde liegenden molekularen Mechanismen von Krankheiten und bestimmten Phänotypen zu gewinnen. Aufgrund der vielfältigen Anwendungsgebiete existieren zahlreiche Tools und Methoden zur Berechnung dieser Ergebnisse. Trotz ihres Nutzens führen bestehende Methoden jedoch oft zu umfangreichen Listen von Gensets, deren inhärente Redundanz die Hypothesenbildung erschwert. Zudem berücksichtigen viele gängige Ansätze zur Auswertung von Functional Enrichment Analysen keine netzwerkbasierten Informationen. Durch Einbeziehung dieser Information könnten Interaktionen zwischen Genset-Mitgliedern besser in Kontext gesetzt und interpretiert werden.

Um diese Herausforderungen anzugehen und die Analyse und Interpretation von Bulk-RNA-seq-Daten zu vereinfachen, haben wir einen standardisierten Workflow veröffentlicht, der reproduzierbare und interaktive Analyseprozesse mit den in unserer Gruppe entwickelten R-Paketen unterstützt [Ludt et al., 2022]. Dieser Workflow führt die Nutzer Schritt für Schritt durch die einzelnen Phasen einer typischen RNA-seq-Analyse, zeigt bewährte Verfahren und fördert die Reproduzierbarkeit. Während der Entwicklung dieses Workflows wurde jedoch ein bedeutendes Defizit offensichtlich: Es fehlte ein speziell entwickeltes Paket zur Vereinfachung der Interpretation von Functional Enrichment Analysen. Unser zuvor entwickeltes Paket `GeneTonic` bietet zwar grundlegende Funktionen zur Untersuchung solcher Ergebnisse, ist aber keineswegs auf diese Aufgabe zugeschnitten. Folglich griffen viele unserer Kooperationspartner weiterhin auf die manuelle Durchsicht der umfangreichen Ergebnislisten zurück, um Muster und interessante Gensets zu identifizieren und diese daraufhin mithilfe von Datenbanken wie der Gene Ontology (GO) oder der Kyoto Encyclopedia of Genes and Genomes (KEGG) weiter zu erforschen. Dieser Prozess birgt jedoch das Risiko einer verzerrten Interpretation, da vertraute und erwartete Gensets leicht erkannt werden, während neue oder unerwartete Informationen möglicherweise übersehen werden – insbesondere aufgrund der großen Anzahl verfügbarer Ergebnisse.

Um zu überprüfen, ob dieses Problem nur in unserer Forschungsgruppe und bei unseren Kooperationspartnern besteht oder ob es sich um ein größeres Problem innerhalb der wissenschaftlichen Gemeinschaft handelt, habe ich eine Literaturrecherche durchgeführt. Diese ergab, dass eine unzureichende Dokumentation und Berichterstattung über die Methoden der Functional Enrichment Analyse weit verbreitet ist. Über 75% der untersuchten Studien gaben nur wenige Details zu ihren Analysen an, was es Fachkollegen schwer bis unmöglich macht, die Ergebnisse zu verifizieren und zu reproduzieren. Außerdem zeigte die Recherche, dass in veröffentlichten Studien häufig Gensets hervorgehoben werden, die aufgrund ihrer statistischen Signifikanz am Anfang der Ergebnisliste stehen. Dies deutet darauf hin, dass die umfangreichen Ergebnislisten nicht immer vollständig untersucht werden, was dazu führen kann, dass wichtige Erkenntnisse verloren gehen.

Im Rahmen dieser Dissertation habe ich daher ein R-Paket entwickelt, das die Interpretation der Ergebnisse der Functional Enrichment Analyse vereinfacht und effizienter gestaltet. Das Ergebnis ist `GeDi`, ein R/Bioconductor-Paket, das Gensets basierend auf verschiedenen Ähnlichkeitsmaßen zu sinnvollen Gruppen (Clustern) zusammenfasst, wodurch Redundanz reduziert und die Übersichtlichkeit der Ergebnisse verbessert wird. `GeDi` erreicht dies durch die Implementierung verschiedener **Ge**nset-**Di**stanzmetriken und Clustering-Algorithmen. Darüber hinaus bezieht `GeDi` Informationen über Protein-Protein-Interaktionen in die Analyse ein, um einen umfassenderen Einblick in die zugrunde liegenden biologischen Prozesse zu geben.

`GeDi` unterstützt eine interaktive Erkundung und detaillierte Analysen der Daten durch eine integrierte `Shiny`-Anwendung. Zusätzlich lässt es sich aber auch nahtlos in bestehende Analyseworkflows integrieren. Dank der flexiblen Anwendungsmöglichkeiten richtet sich `GeDi` an eine breite Nutzergruppe – ein relevanter Aspekt angesichts der zunehmenden Anzahl durchgeführter Functional Enrichment Analysen. Mithilfe von interaktiven Visualisierungen und aggregierter Ergebnisse reduziert `GeDi` nicht nur den Zeitaufwand, der für die Interpretation der Ergebnisse benötigt wird, sondern minimiert auch Verzerrungen, die durch manuelle Auswertung entstehen können. So ermöglicht `GeDi` eine effizientere und objektivere Dateninterpretation, wie in dieser Dissertation an öffentlich zugänglichen Bulk-RNA-seq-Daten gezeigt wurde.

Mit seiner Funktionalität für interaktive Datenerkundung, flexiblen eigenständigen Funktionen und der nahtlosen Integration in unseren standardisierten Bulk-RNA-seq-Workflow hat `GeDi` das Potenzial, die Berichtsstandards für Functional Enrichment Analysen in der wissenschaftlichen Forschung zu verbessern. Darüber hinaus fördert `GeDi` die Reproduzierbarkeit der Analysen durch die automatische Generierung von HTML-Reports.

Durch die Verbesserung der Effizienz, Verfügbarkeit und Reproduzierbarkeit der Dateninterpretation hat `GeDi` das Potenzial, neue Forschungsansätze zu fördern und die Entwicklung neuer Hypothesen zu erleichtern, was letzendlich den Bereich der Omics-Analysen insgesamt voranbringen kann.

Bibliography

Adiconis Xian, Borges-Rivera Diego, Satija Rahul, DeLuca David S, Busby Michele A, Berlin Aaron M, Sivachenko Andrey, Thompson Dawn Anne, Wysoker Alec, Fennell Timothy, Gnirke Andreas, Pochet Nathalie, Regev Aviv, Levin Joshua Z. Comparative analysis of RNA sequencing methods for degraded or low-input samples // Nature Methods. 2013. 10, 7. 623–629.

Ahmad Mubashir, Krüger Benjamin Thilo, Kroll Torsten, Vettorazzi Sabine, Dorn Ann-Kristin, Mengele Florian, Lee Sooyeon, Nandi Sayantan, Yilmaz Dilay, Stolz Miriam, Tangudu Naveen Kumar, Vázquez David Carro, Pachmayr Johanna, Cirstea Ion Cristian, Spasic Maja Vujic, Ploubidou Aspasia, Ignatius Anita, Tuckermann Jan. Inhibition of Cdk5 increases osteoblast differentiation and bone mass and improves fracture healing // Bone Research. 2022. 10, 1. 33.

Akhmedov Murodzhon, Martinelli Axel, Geiger Roger, Kwee Ivo. Omics Playground: A comprehensive self-service platform for visualization, analytics and exploration of Big Omics Data // NAR Genomics and Bioinformatics. 2020. 2, 1. 1–10.

Alberts B. Molecular Biology of the Cell: Hauptbd. 2002.

Alberts B, Johnson A, Lewis J, Morgan D, Raff M, Roberts K, Walter P. Molecular Biology of the Cell. 2014. (500 Tips).

Aleksander Suzi A, Balhoff James, Carbon Seth, Cherry J Michael, Drabkin Harold J, Ebert Dustin, Feuermann Marc, Gaudet Pascale, Harris Nomi L, Hill David P, Lee Raymond, Mi Huaiyu, Moxon Sierra, Mungall Christopher J, Muruganugan Anushya, Mushayahama Tremayne, Sternberg Paul W, Thomas Paul D, Van Auken Kimberly, Ramsey Jolene, Siegele Deborah A, Chisholm Rex L, Fey Petra, Aspromonte Maria Cristina, Nugnes Maria Victoria, Quaglia Federica, Tosatto Silvio, Giglio Michelle, Nadendla Suvarna, Antonazzo Giulia, Attrill Helen, Santos Gil dos, Marygold Steven, Strelets Victor, Tabone Christopher J, Thurmond Jim, Zhou Pinglei, Ahmed Saadullah H, Asanitthong Praoparn, Luna Buitrago Diana, Erdol Meltem N, Gage Matthew C, Ali Kadhum Mohamed, Li Kan Yan Chloe, Long Miao, Michalak Aleksandra, Pesala Angeline, Pritazahra Armalya, Saverimuttu Shirin C C, Su Renzhi, Thurlow Kate E, Lovering Ruth C, Logie Colin, Oliferenko Snezhana, Blake Judith, Christie Karen, Corbani Lori, Dolan Mary E, Drabkin Harold J, Hill David P, Ni Li, Sitnikov Dmitry, Smith Cynthia, Cuzick Alayne, Seager James, Cooper Laurel, Elser Justin, Jaiswal Pankaj, Gupta Parul, Jaiswal Pankaj, Naithani Sushma, Lera-Ramirez Manuel, Rutherford Kim, Wood Valerie, De Pons Jeffrey L, Dwinell Melinda R, Hayman G Thomas, Kaldunski Mary L, Kwitek Anne E, Laulederkind Stanley J F, Tutaj Marek A, Vedi Mahima, Wang Shur-Jen, D'Eustachio Peter, Aimo Lucila, Axelsen Kristian, Bridge Alan, Hyka-Nouspikel Nevila, Morgat Anne, Aleksander Suzi A, Cherry J Michael, Engel Stacia R, Karra Kalpana, Miyasato Stuart R, Nash Robert S, Skrzypek Marek S, Weng Shuai, Wong Edith D, Bakker Erika, Berardini Tanya Z, Reiser Leonore, Auchincloss Andrea, Axelsen Kristian, Argoud-Puy Ghislaine, Blatter Marie-Claude, Boutet Emmanuel, Breuza Lionel, Bridge Alan, Casals-Casas Cristina, Coudert Elisabeth, Estreicher Anne, Livia Famiglietti Maria, Feuermann Marc, Gos Arnaud, Gruaz-Gumowski Nadine, Hulo Chantal, Hyka-Nouspikel Nevila, Jungo Florence, Le Mercier Philippe, Lieberherr Damien, Masson Patrick, Morgat Anne, Pedruzzi Ivo,

Pourcel Lucille, Poux Sylvain, Rivoire Catherine, Sundaram Shyamala, Bateman Alex, Bowler-Barnett Emily, Bye-A-Jee Hema, Denny Paul, Ignatchenko Alexandr, Ishtiaq Rizwan, Lock Antonia, Lussi Yvonne, Magrane Michele, Martin Maria J, Orchard Sandra, Raposo Pedro, Speretta Elena, Tyagi Nidhi, Warner Kate, Zaru Rossana, Diehl Alexander D, Lee Raymond, Chan Juancarlos, Diamantakis Stavros, Raciti Daniela, Zarowiecki Magdalena, Fisher Malcolm, James-Zorn Christina, Ponferrada Virgilio, Zorn Aaron, Ramachandran Sridhar, Ruzicka Leyla, Westerfield Monte. The Gene Ontology knowledgebase in 2023 // Genetics. may 2023. 224, 1. iyad031.

Alexa Adrian, Rahnenführer Jörg, Lengauer Thomas. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure // Bioinformatics. 2006. 22, 13. 1600–1607.

Alie Juhaini, Gustriansyah Rendra. Customer Segmentation For Digital Marketing Based on Shopping Patterns // Jurnal Aplikasi Bisnis dan Manajemen (JABM). jan 2024. 10, 1 SE - Articles. 209.

Almende B.V. and Contributors , Thieurmel Benoit. visNetwork: Network Visualization using 'vis.js' Library. 2022. R package version 2.1.2.

Alwine J C, Kemp D J, Stark G R. Method for detection of specific RNAs in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with DNA probes. // Proceedings of the National Academy of Sciences. dec 1977. 74, 12. 5350–5354.

Amezquita Robert A., Lun Aaron T. L., Becht Etienne, Carey Vince J., Carpp Lindsay N., Geistlinger Ludwig, Marini Federico, Rue-Albrecht Kevin, Risso Davide, Soneson Charlotte, Waldron Levi, Pagès Hervé, Smith Mike L., Huber Wolfgang, Morgan Martin, Gottardo Raphael, Hicks Stephanie C. Orchestrating single-cell analysis with Bioconductor // Nature Methods. feb 2020. 17, 2. 137–145.

Anders Simon, Pyl Paul Theodor, Huber Wolfgang. HTSeq-A Python framework to work with high-throughput sequencing data // Bioinformatics. 2015. 31, 2. 166–169.

Aria Massimo. pubmedR: Gathering Metadata About Publications, Grants, Clinical Trials from 'PubMed' Database. 2020. R package version 0.0.3.

Ashburner Michael, Ball Catherine A, Blake Judith A, Botstein David, Butler Heather, Cherry J Michael, Davis Allan P, Dolinski Kara, Dwight Selina S, Eppig Janan T, Harris Midori A, Hill David P, Issel-Tarver Laurie, Kasarskis Andrew, Lewis Suzanna, Matese John C, Richardson Joel E, Ringwald Martin, Rubin Gerald M, Sherlock Gavin. Gene Ontology: tool for the unification of biology // Nature Genetics. 2000. 25, 1. 25–29.

Baggerly Keith, Berry Donald A. Reproducible Research // AMSTAT News. January 2011. Accessed: 2024-09-12.

Barrett Tanya, Wilhite Stephen E, Ledoux Pierre, Evangelista Carlos, Kim Irene F, Tomashevsky Maxim, Marshall Kimberly A, Phillippy Katherine H, Sherman Patti M, Holko Michelle, Yefanov Andrey, Lee Hyeseung, Zhang Naigong, Robertson Cynthia L, Serova Nadezhda, Davis Sean, Soboleva Alexandra. NCBI GEO: archive for functional genomics data sets—update // Nucleic Acids Research. jan 2013. 41, D1. D991–D995.

Bates Douglas, Maechler Martin, Jagan Mikael. Matrix: Sparse and Dense Matrix Classes and Methods. 2024. R package version 1.7-0.

Battle Leilani, Heer Jeffrey. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau // Computer Graphics Forum. jun 2019. 38, 3. 145–159.

Benjamini Yoav, Hochberg Yosef. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing // Journal of the Royal Statistical Society: Series B (Methodological). jan 1995. 57, 1. 289–300.

Berger Bonnie, Peng Jian, Singh Mona. Computational solutions for omics data // Nature Reviews Genetics. 2013. 14, 5. 333–346.

Blohm Dietmar H, Guiseppi-Elie Anthony. New developments in microarray technology // Current Opinion in Biotechnology. 2001. 12, 1. 41–47.

Blondel Vincent D., Guillaume Jean Loup, Lambiotte Renaud, Lefebvre Etienne. Fast unfolding of communities in large networks // Journal of Statistical Mechanics: Theory and Experiment. 2008. 2008, 10.

Boakye Serebour Tracy, Cribbs Adam P, Baldwin Mathew J, Masimirembwa Collen, Chikwambi Zedias, Kerasidou Angeliki, Snelling Sarah J B. Overcoming barriers to single-cell RNA sequencing adoption in low- and middle-income countries // European Journal of Human Genetics. 2024.

Bray Nicolas L, Pimentel Harold, Melsted Páll, Pachter Lior. Near-optimal probabilistic RNA-seq quantification // Nature Biotechnology. 2016. 34, 5. 525–527.

Britto-Borges Thiago, Ludt Annekathrin, Boileau Etienne, Gjerga Enio, Marini Federico, Dieterich Christoph. Magnetique: an interactive web application to explore transcriptome signatures of heart failure // Journal of Translational Medicine. 2022. 20, 1. 1–9.

Cano-Gamez Eddie, Trynka Gosia. From GWAS to Function: Using Functional Genomics to Identify the Mechanisms Underlying Complex Diseases // Frontiers in Genetics. 2020. 11.

Carbon S., Douglass E., Dunn N., Good B., Harris N. L., Lewis S. E., Mungall C. J., Basu S., Chisholm R. L., Dodson R. J., Hartline E., Fey P., Thomas P. D., Albou L. P., Ebert D., Kesling M. J., Mi H., Muruganujan A., Huang X., Poudel S., Mushayahama T., Hu J. C., LaBonte S. A., Siegele D. A., Antonazzo G., Attrill H., Brown N. H., Fexova S., Garapati P., Jones T. E.M., Marygold S. J., Millburn G. H., Rey A. J., Trovisco V., Dos Santos G., Emmert D. B., Falls K., Zhou P., Goodman J. L., Strelets V. B., Thurmond J., Courtot M., Osumi D. S., Parkinson H., Roncaglia P., Acencio M. L., Kuiper M., Lreid A., Logie C., Lovering R. C., Huntley R. P., Denny P., Campbell N. H., Kramarz B., Acquaah V., Ahmad S. H., Chen H., Rawson J. H., Chibucos M. C., Giglio M., Nadendla S., Tauber R., Duesbury M. J., Del N. T., Meldal B. H.M., Perfetto L., Porras P., Orchard S., Shrivastava A., Xie Z., Chang H. Y., Finn R. D., Mitchell A. L., Rawlings N. D., Richardson L., Sangrador-Vegas A., Blake J. A., Christie K. R., Dolan M. E., Drabkin H. J., Hill D. P., Ni L., Sitnikov D., Harris M. A., Oliver S. G.,

Rutherford K., Wood V., Hayles J., Bahler J., Lock A., Bolton E. R., De Pons J., Dwinell M., Hayman G. T., Laulederkind S. J.F., Shimoyama M., Tutaj M., Wang S. J., D'Eustachio P., Matthews L., Balhoff J. P., Aleksander S. A., Binkley G., Dunn B. L., Cherry J. M., Engel S. R., Gondwe F., Karra K., MacPherson K. A., Miyasato S. R., Nash R. S., Ng P. C., Sheppard T. K., Shrivatsav Vp A., Simison M., Skrzypek M. S., Weng S., Wong E. D., Feuermann M., Gaudet P., Bakker E., Berardini T. Z., Reiser L., Subramaniam S., Huala E., Arighi C., Auchincloss A., Axelsen K., Argoud G. P., Bateman A., Bely B., Blatter M. C., 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 M. J., Masson P., Natale D. A., Hyka N. N., 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 M. C., Howe K., Lee R., Mendel J., Muller H. M., Raciti D., Van Auken K., Berriman M., Stein L., Sternberg P. W., Howe D., Toro S., Westerfield M. The Gene Ontology Resource: 20 years and still GOing strong // Nucleic Acids Research. 2019. 47, D1. D330–D338.

Chang Winston, Borges Ribeiro Barbara. shinydashboard: Create Dashboards with 'Shiny'. 2021. R package version 0.7.2.

Chang Winston, Cheng Joe, Allaire JJ, Sievert Carson, Schloerke Barret, Xie Yihui, Allen Jeff, McPherson Jonathan, Dipert Alan, Borges Barbara. shiny: Web Application Framework for R. 2024. R package version 1.8.1.9000, https://github.com/rstudio/shiny.

Chen Edward Y, Tan Christopher M, Kou Yan, Duan Qiaonan, Wang Zichen, Meirelles Gabriela Vaz, Clark Neil R, Ma'ayan Avi. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool // BMC Bioinformatics. 2013. 14, 1. 128.

Chen Yunshun, Chen Lizhong, Lun Aaron T. L., Baldoni Pedro L., Smyth Gordon K. edgeR 4.0: powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets // bioRxiv. 2024. 2024.01.21.576131.

Chung Neo Christopher, Miasojedow BłaŻej, Startek Michał, Gambin Anna. Jaccard/-Tanimoto similarity test and estimation methods for biological presence-absence data // BMC Bioinformatics. 2019. 20, 15. 644.

Claerbout Jon F, Karrenbach Martin. Electronic documents give reproducible research a new meaning // SEG Technical Program Expanded Abstracts 1992. jan 1992. 601–604. (SEG Technical Program Expanded Abstracts).

Cohen Jacob. A Coefficient of Agreement for Nominal Scales // Educational and Psychological Measurement. 1960. 20, 1. 37–46.

Conesa Ana, Madrigal Pedro, Tarazona Sonia, Gomez-Cabrero David, Cervera Alejandra, McPherson Andrew, Szcześniak Michal Wojciech, Gaffney Daniel J., Elo Laura L., Zhang Xuegong, Mortazavi Ali. A survey of best practices for RNA-seq data analysis // Genome Biology. 2016. 17, 1. 1–19.

Connell Joseph H. Diversity in Tropical Rain Forests and Coral Reefs // Science. 1978. 199, 4335. 1302–1310.

Csárdi Gábor, Nepusz Tamás, Traag Vincent, Horvát Szabolcs, Zanini Fabio, Noom Daniel, Müller Kirill. igraph: Network Analysis and Visualization in R. 2024. R package version 2.0.3.

Dammer Leon, Marini Federico. mosdef: MOSt frequently used and useful Differential Expression Functions. 2024. R package version 1.1.3, commit 5daf035202958b901b5c6eb207b1e183499509dd.

Dayalu Praveen, Albin Roger L. Huntington Disease: Pathogenesis and Treatment // Neurologic Clinics. 2015. 33, 1. 101–114.

Delacher Michael, Imbusch Charles D., Hotz-Wagenblatt Agnes, Mallm Jan Philipp, Bauer Katharina, Simon Malte, Riegel Dania, Rendeiro André F., Bittner Sebastian, Sanderink Lieke, Pant Asmita, Schmidleithner Lisa, Braband Kathrin L., Echtenachter Bernd, Fischer Alexander, Giunchiglia Valentina, Hoffmann Petra, Edinger Matthias, Bock Christoph, Rehli Michael, Brors Benedikt, Schmidl Christian, Feuerer Markus. Precursors for Nonlymphoid-Tissue Treg Cells Reside in Secondary Lymphoid Organs and Are Programmed by the Transcription Factor BATF // Immunity. 2020. 52, 2. 295–312.e11.

Delacher Michael, Imbusch Charles D, Weichenhan Dieter, Breiling Achim, Hotz-Wagenblatt Agnes, Träger Ulrike, Hofer Ann-Cathrin, Kägebein Danny, Wang Qi, Frauhammer Felix, Mallm Jan-Philipp, Bauer Katharina, Herrmann Carl, Lang Philipp A, Brors Benedikt, Plass Christoph, Feuerer Markus. Genome-wide DNA-methylation landscape defines specialization of regulatory T cells in tissues // Nature Immunology. 2017. 18, 10. 1160–1172.

Dice Lee R. Measures of the amount of ecologic association between species // Ecology. 1945. 26, 3. 297–302.

Dillies Marie Agnès, Rau Andrea, Aubert Julie, Hennequet-Antier Christelle, Jeanmougin Marine, Servant Nicolas, Keime Céline, Marot Nicolas Servant, Castel David, Estelle Jordi, Guernec Gregory, Jagla Bernd, Jouneau Luc, Laloë Denis, Le Gall Caroline, Schaëffer Brigitte, Le Crom Stéphane, Guedj Mickaël, Jaffrézic Florence. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis // Briefings in Bioinformatics. 2013. 14, 6. 671–683.

Dobin Alexander, Davis Carrie A., Schlesinger Felix, Drenkow Jorg, Zaleski Chris, Jha Sonali, Batut Philippe, Chaisson Mark, Gingeras Thomas R. STAR: Ultrafast universal RNA-seq aligner // Bioinformatics. 2013. 29, 1. 15–21.

Edgar Ron, Domrachev Michael, Lash Alex E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository // Nucleic Acids Research. 2002. 30, 1. 207–210.

Evans Ciaran, Hardin Johanna, Stoebel Daniel M. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions // Briefings in bioinformatics. 2018. 19, 5. 776–792.

Fabregat Antonio, Jupe Steven, Matthews Lisa, Sidiropoulos Konstantinos, Gillespie Marc, Garapati Phani, Haw Robin, Jassal Bijay, Korninger Florian, May Bruce, Milacic Marija, Roca Corina Duenas, Rothfels Karen, Sevilla Cristoffer, Shamovsky Veronica, Shorser Solomon, Varusai Thawfeek, Viteri Guilherme, Weiser Joel, Wu Guanming, Stein Lincoln, Hermjakob Henning, D'Eustachio Peter. The Reactome Pathway Knowledgebase // Nucleic Acids Research. 2018. 46, D1. D649–D655.

Feinerer Ingo, Hornik Kurt. tm: Text Mining Package. 2024. R package version 0.7-13.

Feinerer Ingo, Hornik Kurt, Meyer David. Text Mining Infrastructure in R // Journal of Statistical Software. March 2008. 25, 5. 1–54.

Fellows Ian. wordcloud: Word Clouds. 2018. R package version 2.6.

Frankish Adam, Diekhans Mark, Ferreira Anne Maud, Johnson Rory, Jungreis Irwin, Loveland Jane, Mudge Jonathan M., Sisu Cristina, Wright James, Armstrong Joel, Barnes If, Berry Andrew, Bignell Alexandra, Carbonell Sala Silvia, Chrast Jacqueline, Cunningham Fiona, Di Domenico Tomás, Donaldson Sarah, Fiddes Ian T., García Girón Carlos, Gonzalez Jose Manuel, Grego Tiago, Hardy Matthew, Hourlier Thibaut, Hunt Toby, Izuogu Osagie G., Lagarde Julien, Martin Fergal J., Martínez Laura, Mohanan Shamika, Muir Paul, Navarro Fabio C.P., Parker Anne, Pei Baikang, Pozo Fernando, Ruffier Magali, Schmitt Bianca M., Stapleton Eloise, Suner Marie Marthe, Sycheva Irina, Uszczynska-Ratajczak Barbara, Xu Jinuri, Yates Andrew, Zerbino Daniel, Zhang Yan, Aken Bronwen, Choudhary Jyoti S., Gerstein Mark, Guigó Roderic, Hubbard Tim J.P., Kellis Manolis, Paten Benedict, Reymond Alexandre, Tress Michael L., Flicek Paul. GENCODE reference annotation for the human and mouse genomes // Nucleic Acids Research. 2019. 47, D1. D766–D773.

Gallery R Graph. Dendrogram with R. 2024a. Accessed: 2024-09-12.

Gallery R Graph. Word Cloud with R. 2024b. Accessed: 2024-09-12.

Gamermann Daniel, Montagud Arnau, Conejero J Alberto, Fernández de Córdoba Pedro, Urchueguía Javier F. Large scale evaluation of differences between network-based and pairwise sequence-alignment-based methods of dendrogram reconstruction // PLOS ONE. sep 2019. 14, 9. e0221631.

Ganz Carl. rintrojs: A Wrapper for the Intro.js Library // Journal of Open Source Software. October 2016. 1.

Garcia-Moreno Adrian, López-Domínguez Raul, Villatoro-García Juan Antonio, Ramirez-Mena Alberto, Aparicio-Puerta Ernesto, Hackenberg Michael, Pascual-Montano Alberto, Carmona-Saez Pedro. Functional Enrichment Analysis of Regulatory Elements. // Biomedicines. mar 2022. 10, 3.

Geistlinger Ludwig, Csaba Gergely, Santarelli Mara, Ramos Marcel, Schiffer Lucas, Turaga Nitesh, Law Charity, Davis Sean, Carey Vincent, Morgan Martin, Zimmer Ralf, Waldron Levi. Toward a gold standard for benchmarking gene set enrichment analysis // Briefings in Bioinformatics. jan 2021. 22, 1. 545–556.

Gentleman Robert C., Carey Vincent J., Bates Douglas M., Bolstad Ben, Dettling Marcel, Dudoit Sandrine, Ellis Byron, Gautier Laurent, Ge Yongchao, Gentry Jeff, Hornik Kurt, Hothorn Torsten, Huber Wolfgang, Iacus Stefano, Irizarry Rafael, Leisch Friedrich, Li Cheng, Maechler Martin, Rossini Anthony J., Sawitzki Gunther, Smith Colin, Smyth Gordon, Tierney Luke, Yang Jean Y.H., Zhang Jianhua. Bioconductor: open software development for computational biology and bioinformatics. // Genome biology. 2004. 5, 10.

Girolami Mark, Mischak Harald, Krebs Ronald. Analysis of complex, multidimensional datasets // Drug Discovery Today: Technologies. 2006. 3, 1. 13–19.

Gohlke Julia M, Thomas Reuben, Zhang Yonqing, Rosenstein Michael C, Davis Allan P, Murphy Cynthia, Becker Kevin G, Mattingly Carolyn J, Portier Christopher J. Genetic and environmental pathways to complex diseases // BMC Systems Biology. 2009. 3, 1. 46.

Goldacre Ben, Morton Caroline E, DeVito Nicholas J. Why researchers should share their analytic code // BMJ. nov 2019. 367. l6365.

Granjon David. bs4Dash: A 'Bootstrap 4' Version of 'shinydashboard'. 2024. R package version 2.3.3, https://github.com/RinteRface/bs4Dash.

Graves David J. Powerful tools for genetic analysis come of age // Trends in Biotechnology. 1999. 17, 3. 127–134.

Gu Zuguang. Complex Heatmap Visualization // iMeta. 2022.

Gu Zuguang, Eils Roland, Schlesner Matthias. Complex heatmaps reveal patterns and correlations in multidimensional genomic data // Bioinformatics. 2016.

He Qingmin, Liu Chuan, Wang Xiaohan, Rong Kang, Zhu Mingyang, Duan Liying, Zheng Pengyuan, Mi Yang. Exploring the mechanism of curcumin in the treatment of colon cancer based on network pharmacology and molecular docking // Frontiers in Pharmacology. 2023. 14.

Hellerstein J.M., Avnur R., Chou A., Hidber C., Olston C., Raman V., Roth T., Haas P.J. Interactive data analysis: the Control project // Computer. 1999. 32, 8. 51–59.

Heumos Lukas, Schaar Anna C, Lance Christopher, Litinetskaya Anastasia, Drost Felix, Zappia Luke, Lücken Malte D, Strobl Daniel C, Henao Juan, Curion Fabiola, Aliee Hananeh, Ansari Meshal, Mompel Pau Badia-i, Büttner Maren, Dann Emma, Dimitrov Daniel, Dony Leander, Frishberg Amit, He Dongze, Hediyeh-zadeh Soroor, Hetzel Leon, Ibarra Ignacio L, Jones Matthew G, Lotfollahi Mohammad, Martens Laura D, Müller Christian L, Nitzan Mor, Ostner Johannes, Palla Giovanni, Patro Rob, Piran Zoe, Ramírez-Suástegui Ciro, Saez-Rodriguez Julio, Sarkar Hirak, Schubert Benjamin, Sikkema Lisa, Srivastava Avi, Tanevski Jovan, Virshup Isaac, Weiler Philipp, Schiller Herbert B, Theis Fabian J, Consortium Single-cell Best Practices. Best practices for single-cell analysis across modalities // Nature Reviews Genetics. 2023. 24, 8. 550–572.

Hong Mingye, Tao Shuang, Zhang Ling, Diao Li Ting, Huang Xuanmei, Huang Shaohui, Xie Shu Juan, Xiao Zhen Dong, Zhang Hua. RNA sequencing: new technologies and applications in cancer research // Journal of Hematology and Oncology. 2020. 13, 1. 1–16.

Huang Da Wei, Sherman Brad T, Lempicki Richard A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. // Nucleic acids research. jan 2009. 37, 1. 1–13.

Huang Da Wei, Sherman Brad T, Tan Qina, Collins Jack R, Alvord W Gregory, Roayaei Jean, Stephens Robert, Baseler Michael W, Lane H Clifford, Lempicki Richard A. The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists // Genome Biology. 2007. 8, 9. R183.

Huang Dawei, Zhong Guixian, Zhang Shiyang, Jiang Kerui, Wang Chen, Wu Jian, Wang Bo. Trichome-Specific Analysis and Weighted Gene Co-Expression Correlation Network Analysis (WGCNA) Reveal Potential Regulation Mechanism of Artemisinin Biosynthesis in Artemisia annua // International Journal of Molecular Sciences. 2023. 24, 10.

Huber Wolfgang, Carey Vincent J, Gentleman Robert, Anders Simon, Carlson Marc, Carvalho Benilton S, Bravo Hector Corrada, Davis Sean, Gatto Laurent, Girke Thomas, Gottardo Raphael, Hahne Florian, Hansen Kasper D, Irizarry Rafael A, Lawrence Michael, Love Michael I, MacDonald James, Obenchain Valerie, Oleś Andrzej K, Pagès Hervé, Reyes Alejandro, Shannon Paul, Smyth Gordon K, Tenenbaum Dan, Waldron Levi, Morgan Martin. Orchestrating high-throughput genomic analysis with Bioconductor // Nature Methods. 2015. 12, 2. 115–121.

Huttlin Edward L, Bruckner Raphael J, Paulo Joao A, Cannon Joe R, Ting Lily, Baltier Kurt, Colby Greg, Gebreab Fana, Gygi Melanie P, Parzen Hannah, Szpyt John, Tam Stanley, Zarraga Gabriela, Pontano-Vaites Laura, Swarup Sharan, White Anne E, Schweppe Devin K, Rad Ramin, Erickson Brian K, Obar Robert A, Guruharsha K G, Li Kejie, Artavanis-Tsakonas Spyros, Gygi Steven P, Harper J Wade. Architecture of the human interactome defines protein communities and disease networks // Nature. 2017. 545, 7655. 505–509.

Jaccard Paul. the Distribution of the Flora in the Alpine Zone. // New Phytologist. 1912. 11, 2. 37–50.

Jiang Gao, Zheng Juan Yu, Ren Shu Ning, Yin Weilun, Xia Xinli, Li Yun, Wang Hou Ling. A comprehensive workflow for optimizing RNA-seq data analysis // BMC Genomics. 2024. 25, 1. 1–21.

Jiang Jay J., Conrath David W. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy // Proceedings of the 10th Research on Computational Linguistics International Conference. Taipei, Taiwan: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), VIII 1997. 19–33.

Jollife Ian T., Cadima Jorge. Principal component analysis: A review and recent developments // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2016. 374, 2065.

Jolliffe Ian T. Principal Component Analysis. 30, 3. 2002. Second Edition. 487. (Encyclopedia of Statistics in Behavioral Science).

Jones Piet, Weighill Deborah, Shah Manesh, Climer Sharlee, Schmutz Jeremy, Sreedasyam Avinash, Tuskan Gerald, Jacobson Daniel. Network Modeling of Complex Data Sets // Metabolic Pathway Engineering. New York, NY: Springer US, 2020. 197–215.

Kanehisa Minoru, Furumichi Miho, Tanabe Mao, Sato Yoko, Morishima Kanae. KEGG: New perspectives on genomes, pathways, diseases and drugs // Nucleic Acids Research. 2017. 45, D1. D353–D361.

Kanehisa Minoru, Sato Yoko, Furumichi Miho, Morishima Kanae, Tanabe Mao. New approach for understanding genome variations in KEGG // Nucleic Acids Research. 2019. 47, D1. D590–D595.

Kaufman L., Rousseeuw P. Clustering by means of Medoids. 1987. 405–416.

Keim D A, Mansmann F, Schneidewind J, Ziegler H. Challenges in Visual Data Analysis // Tenth International Conference on Information Visualisation (IV'06). 2006. 9–16.

Khatri Purvesh, Sirota Marina, Butte Atul J. Ten years of pathway analysis: Current approaches and outstanding challenges // PLoS Computational Biology. feb 2012. 8, 2. e1002375.

Kim Daehwan, Paggi Joseph M, Park Chanhee, Bennett Christopher, Salzberg Steven L. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype // Nature Biotechnology. 2019. 37, 8. 907–915.

Kim Sangkyu, Fuselier Jessica, Latoff Anna, Manges Justin, Jazwinski S Michal, Zsombok Andrea. Upregulation of extracellular proteins in a mouse model of Alzheimer's disease // Scientific Reports. 2023. 13, 1. 6998.

Kodali Maheedhar, Madhu Leelavathi N, Reger Roxanne L, Milutinovic Bojana, Upadhya Raghavendra, Gonzalez Jenny J, Attaluri Sahithi, Shuai Bing, Gitai Daniel L G, Rao Shama, Choi Jong M, Jung Sung Y, Shetty Ashok K. Intranasally administered human MSC-derived extracellular vesicles inhibit NLRP3-p38/MAPK signaling after TBI and prevent chronic brain dysfunction // Brain, Behavior, and Immunity. 2023. 108. 118–134.

Krebs J E, Goldstein E S, Kilpatrick S T. Lewin's Genes Twelve. 2017.

Kukurba K. R., Montgomery S. B. RNA Sequencing and Analysis // Cold Spring Harbor Protocols. IV 2015. 2015, 11. 951–969.

Kuleshov Maxim V., Jones Matthew R., Rouillard Andrew D., Fernandez Nicolas F., Duan Qiaonan, Wang Zichen, Koplev Simon, Jenkins Sherry L., Jagodnik Kathleen M., Lachmann Alexander, McDermott Michael G., Monteiro Caroline D., Gundersen Gregory W., Maayan Avi. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update // Nucleic Acids Research. 2016. 44, 1. W90–W97.

Lähnemann David, Köster Johannes, Szczurek Ewa, McCarthy Davis J, Hicks Stephanie C, Robinson Mark D, Vallejos Catalina A, Campbell Kieran R, Beerenwinkel Niko, Mahfouz Ahmed, Pinello Luca, Skums Pavel, Stamatakis Alexandros, Attolini Camille Stephan-Otto, Aparicio Samuel, Baaijens Jasmijn, Balvert Marleen, Barbanson Buys de, Cappuccio Antonio, Corleone Giacomo, Dutilh Bas E, Florescu Maria, Guryev Victor, Holmer Rens, Jahn Katharina, Lobo Thamar Jessurun, Keizer Emma M, Khatri Indu, Kielbasa Szymon M, Korbel Jan O, Kozlov Alexey M, Kuo Tzu-Hao, Lelieveldt Boudewijn P F, Mandoiu Ion I, Marioni John C, Marschall Tobias, Mölder Felix, Niknejad Amir, Rączkowska Alicja, Reinders Marcel, Ridder Jeroen de, Saliba Antoine-Emmanuel, Somarakis Antonios, Stegle Oliver, Theis Fabian J, Yang Huan, Zelikovsky Alex, McHardy Alice C, Raphael Benjamin J, Shah Sohrab P, Schönhuth Alexander. Eleven grand challenges in single-cell data science // Genome Biology. 2020. 21, 1. 31.

Lee Eun Ji, Suh Minseok, Choi Hongyoon, Choi Yoori, Hwang Do Won, Bae Sungwoo, Lee Dong Soo. Spatial transcriptomic brain imaging reveals the effects of immunomodulation therapy on specific regional brain cells in a mouse dementia model // BMC Genomics. 2024. 25, 1. 516.

Levandowsky Michael, Winter David. Distance between sets [5] // Nature. 1971. 234, 5323. 34–35.

Li Bo, Dewey Colin N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome // BMC Bioinformatics. 2011. 12, 1. 323.

Li Xinmin, Wang Cun-Yu. From bulk, single-cell to spatial RNA sequencing // International Journal of Oral Science. 2021. 13, 1. 36.

Liao Yang, Smyth Gordon K., Shi Wei. The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads // Nucleic Acids Research. 2019. 47. e47.

Liberzon Arthur, Birger Chet, Thorvaldsdóttir Helga, Ghandi Mahmoud, Mesirov Jill P., Tamayo Pablo. The Molecular Signatures Database Hallmark Gene Set Collection // Cell Systems. 2015. 1, 6. 417–425.

Liberzon Arthur, Subramanian Aravind, Pinchback Reid, Thorvaldsdóttir Helga, Tamayo Pablo, Mesirov Jill P. Molecular signatures database (MSigDB) 3.0 // Bioinformatics. 2011. 27, 12. 1739–1740.

Lin Dekang. An Information-Theoretic Definition of Similarity // Proceedings of the Fifteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. 296–304. (ICML '98).

Lloyd S. Least squares quantization in PCM // IEEE Transactions on Information Theory. 1982. 28, 2. 129–137.

Love Michael I., Anders Simon, Kim Vladislav, Huber Wolfgang. RNA-Seq workflow: gene-level exploratory analysis and differential expression // F1000Research. 2015. 4. 1070.

Love Michael I., Huber Wolfgang, Anders Simon. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 // Genome Biology. 2014. 15. 550.

Ludt Annekathrin, Ustjanzew Arsenij, Binder Harald, Strauch Konstantin, Marini Federico. Interactive and Reproducible Workflows for Exploring and Modeling RNA-seq Data with pcaExplorer, Ideal, and GeneTonic // Current Protocols. 2022. 2, 4. 1–55.

Ma Jing, Shojaie Ali, Michailidis George. A comparative study of topology-based pathway enrichment analysis methods // BMC Bioinformatics. 2019. 20, 1. 546.

Ma Tao, Guo Lingyun, Yan Huihuang, Wang Liguo. Cobind: quantitative analysis of the genomic overlaps // Bioinformatics Advances. jan 2023. 3, 1. vbad104.

Maechler Martin, Rousseeuw Peter, Struyf Anja, Hubert Mia, Hornik Kurt. cluster: Cluster Analysis Basics and Extensions. 2023. R package version 2.1.6 — For new features, see the 'NEWS' and the 'Changelog' file in the package source).

Marini Federico, Binder Harald. pcaExplorer: an R/Bioconductor package for interacting with RNA-seq principal components // BMC Bioinformatics. Jun 2019. 20, 1. 331.

Marini Federico, Linke Jan, Binder Harald. ideal: an R/Bioconductor package for Interactive Differential Expression Analysis // BMC Bioinformatics. 2020. 21. 565.

Marini Federico, Ludt Annekathrin, Linke Jan, Strauch Konstantin. GeneTonic: an R/Bioconductor package for streamlining the interpretation of RNA-seq data // BMC Bioinformatics. 2021. 22, 1. 1–19.

Markowetz Florian. Five selfish reasons to work reproducibly // Genome Biology. 2015. 16, 1. 274.

McKiernan Erin C, Bourne Philip E, Brown C Titus, Buck Stuart, Kenall Amye, Lin Jennifer, McDougall Damon, Nosek Brian A, Ram Karthik, Soderberg Courtney K, Spies Jeffrey R, Thaney Kaitlin, Updegrove Andrew, Woo Kara H, Yarkoni Tal. How open science helps researchers succeed // eLife. 2016. 5. e16800.

Mead A. Review of the Development of Multidimensional Scaling Methods // Journal of the Royal Statistical Society. Series D (The Statistician). sep 1992. 41, 1. 27–39.

Merico Daniele, Isserlin Ruth, Stueker Oliver, Emili Andrew, Bader Gary D. Enrichment map: A network-based method for gene-set enrichment visualization and interpretation // PLoS ONE. 2010. 5, 11.

Miyakawa Tsuyoshi. No raw data, no science: another possible source of the reproducibility crisis // Molecular Brain. 2020. 13, 1. 24.

Monga Isha, Kaur Karambir, Dhanda Sandeep Kumar. Revisiting hematopoiesis: applications of the bulk and single-cell transcriptomics dissecting transcriptional heterogeneity in hematopoietic stem cells // Briefings in Functional Genomics. may 2022. 21, 3. 159–176.

Morgan Martin, Wang Jiefei, Obenchain Valerie, Lang Michel, Thompson Ryan, Turaga Nitesh. BiocParallel: Bioconductor facilities for parallel evaluation. 2024. R package version 1.38.0.

Mortazavi Ali, Williams Brian A, McCue Kenneth, Schaeffer Lorian, Wold Barbara. Mapping and quantifying mammalian transcriptomes by RNA-Seq // Nature Methods. 2008. 5, 7. 621–628.

Munafò Marcus R, Chambers Chris, Collins Alexandra, Fortunato Laura, Macleod Malcolm. The reproducibility debate is an opportunity, not a crisis // BMC Research Notes. 2022. 15, 1. 43.

Munafò Marcus R, Nosek Brian A, Bishop Dorothy V M, Button Katherine S, Chambers Christopher D, Percie du Sert Nathalie, Simonsohn Uri, Wagenmakers Eric-Jan, Ware Jennifer J, Ioannidis John P A. A manifesto for reproducible science // Nature Human Behaviour. 2017. 1, 1. 21.

Mushtaq Hassan, Khawaja Sajid G, Akram Muhammad U, Yasin Amanullah, Muzammal Muhammad, Khalid Shehzad, Khan Shoab A. A Parallel Architecture for the Partitioning around Medoids (PAM) Algorithm for Scalable Multi-Core Processor Implementation with Applications in Healthcare. 2018.

National Heart, Lung, and Blood Institute . Sickle Cell Disease. 2024. Accessed: 2024-10-07.

Olechnowicz Anna, Blatkiewicz Małgorzata, Jopek Karol, Isalan Mark, Mielcarek Michal, Rucinski Marcin. Deregulated Transcriptome as a Platform for Adrenal Huntington's Disease-Related Pathology. 2024.

Oughtred Rose, Rust Jennifer, Chang Christie, Breitkreutz Bobby-Joe, Stark Chris, Willems Andrew, Boucher Lorrie, Leung Genie, Kolas Nadine, Zhang Frederick, Dolma Sonam, Coulombe-Huntington Jasmin, Chatr-aryamontri Andrew, Dolinski Kara, Tyers Mike. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions // Protein Science. jan 2021. 30, 1. 187–200.

Page Matthew J, Nguyen Phi-Yen, Hamilton Daniel G, Haddaway Neal R, Kanukula Raju, Moher David, McKenzie Joanne E. Data and code availability statements in systematic reviews of interventions were often missing or inaccurate: a content analysis // Journal of Clinical Epidemiology. 2022. 147. 1–10.

Paradis Emmanuel, Claude Julien, Strimmer Korbinian. APE: Analyses of Phylogenetics and Evolution in R language // Bioinformatics. jan 2004. 20, 2. 289–290.

Patro Rob, Duggal Geet, Love Michael I., Irizarry Rafael A., Kingsford Carl. Salmon provides fast and bias-aware quantification of transcript expression // Nature Methods. 2017. 14, 4. 417–419.

Pertea Mihaela, Kim Daehwan, Pertea Geo M, Leek Jeffrey T, Salzberg Steven L. Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown // Nature Protocols. 2016. 11, 9. 1650–1667.

Pilz Robin A, Skowronek Dariush, Mellinger Lara, Bekeschus Sander, Felbor Ute, Rath Matthias. Endothelial Differentiation of CCM1 Knockout iPSCs Triggers the Establishment of a Specific Gene Expression Signature. 2023.

R Core Team . R: A Language and Environment for Statistical Computing. Vienna, Austria, 2024.

Raghavachari Nalini, Garcia-Reyero Natàlia. Gene Expression Analysis Methods and Protocols Methods in Molecular Biology 1783. 2018. 171–183.

Reimand Jüri, Isserlin Ruth, Voisin Veronique, Kucera Mike, Tannus-Lopes Christian, Rostamianfar Asha, Wadi Lina, Meyer Mona, Wong Jeff, Xu Changjiang, Merico Daniele, Bader Gary D. Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap // Nature Protocols. 2019. 14, 2. 482–517.

Resnik P. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language // Journal of Artificial Intelligence Research. VII 1999. 11. 95–130.

Ritchie Matthew E, Phipson Belinda, Wu Di, Hu Yifang, Law Charity W, Shi Wei, Smyth Gordon K. limma powers differential expression analyses for RNA-sequencing and microarray studies // Nucleic Acids Research. 2015. 43, 7. e47.

Robinson Mark D, McCarthy Davis J, Smyth Gordon K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data // Bioinformatics. 2010. 26, 1. 139–140.

Rue-Albrecht Kevin, Marini Federico, Soneson Charlotte, Lun Aaron T. L. iSEE: Interactive SummarizedExperiment Explorer // F1000Research. Jun 2018. 7. 741.

Rychlý Pavel. A Lexicographer-Friendly Association Score // Proceedings of the Second Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2008). 2008. 6–9.

Sakaguchi Shimon, Yamaguchi Tomoyuki, Nomura Takashi, Ono Masahiro. Regulatory T Cells and Immune Tolerance // Cell. 2008. 133, 5. 775–787.

Schena Mark, Shalon Dari, Davis Ronald W, Brown Patrick O. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray // Science. oct 1995. 270, 5235. 467–470.

Schlicker Andreas, Domingues Francisco S, Rahnenführer Jörg, Lengauer Thomas. A new measure for functional similarity of gene products based on Gene Ontology // BMC Bioinformatics. 2006. 7, 1. 302.

Schurch Nicholas J., Schofield Pietá, Gierliński Marek, Cole Christian, Sherstnev Alexander, Singh Vijender, Wrobel Nicola, Gharbi Karim, Simpson Gordon G., Owen-Hughes Tom, Blaxter Mark, Barton Geoffrey J. How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? // Rna. 2016. 22, 6. 839–851.

Seal Ruth L., Braschi Bryony, Gray Kristian, Jones Tamsin E.M., Tweedie Susan, Haim-Vilmovsky Liora, Bruford Elspeth A. Genenames.org: the HGNC resources in 2023 // Nucleic acids research. 2023. 51, D1. D1003–D1009.

Seth Soumita, Mallik Saurav, Bhadra Tapas, Zhao Zhongming. Dimensionality Reduction and Louvain Agglomerative Hierarchical Clustering for Cluster-Specified Frequent Biomarker Discovery in Single-Cell Sequencing Data // Frontiers in Genetics. 2022. 13.

Sievert Carson. Interactive Web-Based Data Visualization with R, plotly, and shiny. 2020.

Soneson Charlotte, Delorenzi Mauro. A comparison of methods for differential expression analysis of RNA-seq data // BMC Bioinformatics. 2013. 14, 1. 91.

Soneson Charlotte, Love Michael I., Robinson Mark D. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences // F1000Research. 2015. 4.

Sørensen T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons // Kongelige Danske Videnskabernes Selskab. 1948. 5. 1–34.

Stark Rory, Grzelak Marta, Hadfield James. RNA sequencing: the teenage years // Nature Reviews Genetics. 2019. 20, 11. 631–656.

Stodden Victoria, Borwein Jonathan, Bailey David H. Setting the default to reproducible // Computational Science Research. SIAM News. 2013a. 46. 4–6.

Stodden Victoria, Guo Peixuan, Ma Zhaokun. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals // PLOS ONE. jun 2013b. 8, 6. e67111.

Stuart Tim, Satija Rahul. Integrative single-cell analysis // Nature Reviews Genetics. 2019. 20, 5. 257–272.

Subramanian Aravind, Tamayo Pablo, Mootha Vamsi K., Mukherjee Sayan, Ebert Benjamin L., Gillette Michael A., Paulovich Amanda, Pomeroy Scott L., Golub Todd R., Lander Eric S., Mesirov Jill P. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles // Proceedings of the National Academy of Sciences of the United States of America. 2005. 102, 43. 15545–15550.

Supek Fran, Bošnjak Matko, Škunca Nives, Šmuc Tomislav. Revigo summarizes and visualizes long lists of gene ontology terms // PLoS ONE. 2011. 6, 7.

Supek Fran, Škunca Nives. Visualizing GO annotations // Methods in Molecular Biology. 2017. 1446, 1. 207–220.

Szklarczyk Damian, Kirsch Rebecca, Koutrouli Mikaela, Nastou Katerina, Mehryary Farrokh, Hachilif Radja, Gable Annika L., Fang Tao, Doncheva Nadezhda T., Pyysalo Sampo, Bork Peer, Jensen Lars J., Von Mering Christian. The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest // Nucleic Acids Research. 2023. 51, 1 D. D638–D646.

The ENCODE Project Consortium . A User's Guide to the Encyclopedia of DNA Elements (ENCODE) // PLOS Biology. 2011. 9, 4. 1–21.

The Galaxy Community . The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update // Nucleic Acids Research. jul 2024. 52, W1. W83–W94.

Trapnell Cole, Pachter Lior, Salzberg Steven L. TopHat: Discovering splice junctions with RNA-Seq // Bioinformatics. 2009. 25, 9. 1105–1111.

Trapnell Cole, Roberts Adam, Goff Loyal, Pertea Geo, Kim Daehwan, Kelley David R, Pimentel Harold, Salzberg Steven L, Rinn John L, Pachter Lior. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks // Nature Protocols. 2012. 7, 3. 562–578.

Trayhurn Paul. Northern blotting // Proceedings of the Nutrition Society. 1996. 55, 1B. 583–589.

Ushey Kevin, Wickham Hadley. renv: Project Environments. 2024. R package version 1.0.11.

Van Den Berge Koen, Hembach Katharina M., Soneson Charlotte, Tiberi Simone, Clement Lieven, Love Michael I., Patro Rob, Robinson Mark D. RNA Sequencing Data: Hitchhiker's Guide to Expression Analysis // Annual Review of Biomedical Data Science. 2019. 2. 139–173.

Vijaymeena M.K, Kavitha K. A Survey on Similarity Measures in Text Mining // Machine Learning and Applications: An International Journal (MLAIJ) Vol.3, No.1. 2016.

Wald Abraham. Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations is Large // Transactions of the American Mathematical Society. oct 1943. 54, 3. 426–482.

Wang James Z, Du Zhidian, Payattakool Rapeeporn, Yu Philip S, Chen Chin-Fu. A new method to measure the semantic similarity of GO terms // Bioinformatics. may 2007. 23, 10. 1274–1281.

Wang Zhong, Gerstein Mark, Snyder Michael. Nrg2484-1 // NATURE REVIEwS | genetics. 2009. VOLUME 10, jANUARy 2009. 57–63.

Watson James D, Crick Francis HC, others . Molecular structure of nucleic acids // Nature. 1953. 171, 4356. 737–738.

Wickham Hadley. Mastering Shiny: Build Interactive Apps, Reports, and Dashboards Powered by R. Sebastopol, CA: O'Reilly Media, 2021.

Wickham Hadley, Bryan Jennifer. readxl: Read Excel Files. 2023. R package version 1.4.3.

Wijesooriya Kaumadi, Jadaan Sameer A., Perera Kaushalya L., Kaur Tanuveer, Ziemann Mark. Urgent need for consistent standards in functional enrichment analysis // PLoS Computational Biology. 2022. 18, 3. 1–14.

Winkler Hans. Verbreitung und Ursache der Parthenogenesis im Pflanzen- und Tierreiche. Jena, Germany: Gustav Fischer, 1920.

Wu Tianzhi, Hu Erqiang, Xu Shuangbin, Chen Meijun, Guo Pingfan, Dai Zehan, Feng Tingze, Zhou Lang, Tang Wenli, Zhan Li, Fu Xiaocong, Liu Shanshan, Bo Xiaochen, Yu Guangchuang. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. // Innovation (Cambridge (Mass.)). aug 2021. 2, 3. 100141.

Xu Xuemei, Zhang Xiang, Zhang Guoying, Abbasi Tadi Danyal. Prevalence of antibiotic resistance of Staphylococcus aureus in cystic fibrosis infection: a systematic review and meta-analysis // Journal of Global Antimicrobial Resistance. 2024. 36. 419–425.

Yang In Seok, Kim Sangwoo. Analysis of Whole Transcriptome Sequencing Data: Workflow and Software // Genomics & Informatics. 2015. 13, 4. 119.

Yates Andrew D., Achuthan Premanand, Akanni Wasiu, Allen James, Allen Jamie, Alvarez-Jarreta Jorge, Amode M. Ridwan, Armean Irina M., Azov Andrey G., Bennett Ruth, Bhai Jyothish, Billis Konstantinos, Boddu Sanjay, Marugán José Carlos, Cummins Carla, Davidson Claire, Dodiya Kamalkumar, Fatima Reham, Gall Astrid, Giron Carlos Garcia, Gil Laurent, Grego Tiago, Haggerty Leanne, Haskell Erin, Hourlier Thibaut, Izuogu Osagie G., Janacek Sophie H., Juettemann Thomas, Kay Mike, Lavidas Ilias, Le Tuan, Lemos Diana, Martinez Jose Gonzalez, Maurel Thomas, McDowall Mark, McMahon Aoife, Mohanan Shamika, Moore Benjamin, Nuhn Michael, Oheh Denye N., Parker Anne, Parton Andrew, Patricio Mateus, Sakthivel Manoj Pandian, Abdul Salam Ahamed Imran, Schmitt Bianca M., Schuilenburg Helen, Sheppard Dan, Sycheva Mira, Szuba Marek, Taylor Kieron, Thormann Anja, Threadgold Glen, Vullo Alessandro, Walts Brandon, Winterbottom Andrea, Zadissa Amonida, Chakiachvili Marc, Flint Bethany, Frankish Adam, Hunt Sarah E., Iisley Garth, Kostadima Myrto, Langridge Nick, Loveland Jane E., Martin Fergal J., Morales Joannella, Mudge Jonathan M., Muffato Matthieu, Perry Emily, Ruffier Magali, Trevanion Stephen J., Cunningham Fiona, Howe Kevin L., Zerbino Daniel R., Flicek Paul. Ensembl 2020 // Nucleic Acids Research. 2020. 48, D1. D682–D688.

Yi Haidong, Plotkin Alec, Stanley Natalie. Benchmarking differential abundance methods for finding condition-specific prototypical cells in multi-sample single-cell datasets // Genome Biology. 2024. 25, 1. 9.

Yoon Sora, Kim Jinhwan, Kim Seon-Kyu, Baik Bukyung, Chi Sang-Mun, Kim Seon-Young, Nam Dougu. GScluster: network-weighted gene-set clustering analysis // BMC Genomics. 2019. 20, 1. 352.

Yu Guangchuang, Li Fei, Qin Yide, Bo Xiaochen, Wu Yibo, Wang Shengqi. GOSemSim: An R package for measuring semantic similarity among GO terms and gene products // Bioinformatics. 2010. 26, 7. 976–978.

Zeng Xiaoyu, Shi Gaoli, He Qiankun, Zhu Pingping. Screening and predicted value of potential biomarkers for breast cancer using bioinformatics analysis // Scientific Reports. 2021. 11, 1. 20799.

Zijdenbos A.P., Dawant B.M., Margolin R.A., Palmer A.C. Morphometric analysis of white matter lesions in MR images: method and validation // IEEE Transactions on Medical Imaging. 1994. 13, 4. 716–724.

de Vries Andrie, Ripley Brian D. ggdendro: Create Dendrograms and Tree Diagrams Using 'ggplot2'. 2024. R package version 0.2.0.

del Toro Noemi, Shrivastava Anjali, Ragueneau Eliot, Meldal Birgit, Combe Colin, Barrera Elisabet, Perfetto Livia, How Karyn, Ratan Prashansa, Shirodkar Gautam, Lu Odilia, Mészáros Bálint, Watkins Xavier, Pundir Sangya, Licata Luana, Iannuccelli Marta, Pellegrini Matteo, Martin Maria Jesus, Panni Simona, Duesbury Margaret, Vallet Sylvain D, Rappsilber Juri, Ricard-Blum Sylvie, Cesareni Gianni, Salwinski Lukasz, Orchard Sandra, Porras Pablo, Panneerselvam Kalpana, Hermjakob Henning. The IntAct database: efficient access to fine-grained molecular interaction data // Nucleic Acids Research. jan 2022. 50, D1. D648–D653.

van Dongen S. A cluster algorithm for graphs. 2000.

List of Figures

Appendix

Appendix A: List of Data and Code Availability

The following table provides a comprehensive list of all external links referenced throughout this thesis, along with a brief description of the content each link provides. These links include repositories, datasets, software packages, and other relevant resources that support the methodologies and findings discussed in the thesis. The links are ordered according to their appearance in the thesis to maintain a logical flow and easy reference.

Number	Link	Description
1	`https://annekathrinsilvia.github.io/GeDi/reference/index.html`	A link to the comprehensive documentation of `GeDi`'s functionality, which includes detailed documentation of the available functions and function arguments of the package.
2	`https://www.biorender.com/`	Biorender is a web-based design tool which is frequently used to create scientific illustrations and diagrams. Biorender possesses a vast library of pre-made icons and templates tailored to life science and biological processes.
3	`https://cran.r-project.org`	The Comprehensive R Archive Network (CRAN) is the official repository for R packages, offering a vast collection of software tools for data analysis, visualisation, and statistical computing. It also provides documentation, manuals, and resources for R users.
4	`https://www.bioconductor.org/`	Bioconductor is a repository of R packages focused on bioinformatics and computational biology. It includes packages for genomic data analysis, integration, and visualisation, supporting high-throughput sequencing, microarrays, and other data types.

5	https://github.com/	GitHub is a platform for hosting and sharing code repositories, supporting collaboration on software development projects. It provides tools for version control, issue tracking, and project management, making it a key resource for open-source software development.
6	https://bioconductor.org/packages/GeDi	The Bioconductor page for the GeDi package, providing a summary of the package as well as installation commands.
7	https://github.com/AnnekathrinSilvia/GeDi	The GitHub repository for the GeDi package, containing source code, issue tracking, and installation instructions for developers and users.
8	https://pubmed.ncbi.nlm.nih.gov/	PubMed is a comprehensive database of biomedical literature, providing access to millions of research articles, abstracts, and references. In this thesis, Pubmed was used for a manual literature search.
9	https://github.com/imbeimainz/HIPSTER	The GitHub repository corresponding to the literature review of this thesis. The repository contains the results of the manual Pubmed search, as well as the R code of the literature searches using the pubmedR package. Moreover, the repository includes the seed as well as the code for the random sampling of articles.
10	https://www.bioconductor.org/packages/release/data/annotation/	A link to the annotation packages available in Bioconductor. These packages provide information to annotate the raw experimental data at hand with additional biological, functional, structural or genomic context.
11	http://shiny.imbei.uni-mainz.de:3838/GeDi/	The Shiny demo server of GeDi, which allows users to explore and use the application.

12	https://posit.co/products/open-source/shiny-server/	An information page on Shiny Server, which is an open-source software for hosting and deploying interactive web applications built with R.
13	https://www.shinyapps.io/	A platform for deploying and sharing Shiny web applications, which enables users to host interactive data visualisations and tools online.
14	https://github.com/AnnekathrinSilvia/GSE130842_Showcase	The GitHub repository containing the analysis and preparation code used to analyse the Treg dataset in this thesis. The repository also contains the used input data, as well as the code for some of the figures included in this thesis.
15	https://doi.org/10.5281/zenodo.13843024	The link to the Zenodo repository containing the full evaluation of the reviewed articles used in the literature review of this thesis.
16	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE130842	The GEO entry GSE130842 of the Treg dataset, which was used for the showcase in this thesis. Under this link, the transcriptome data, sample information, as well as the downloadable data files can be found.
17	https://github.com/lottawagner/NetworkHub	The GitHub repository of NetworkHub, an R/Bioconductor package currently under development. Once completed, NetworkHub will provide functionality for retrieving, caching, processing, and preparing PPI networks and their associated information from multiple databases.

Overview of the External Links

Appendix B: List of Reviewed Articles

20 Randomly Selected Articles from Manual PubMed Search

1. Mi, M. Y., Barber, J. L., Rao, P., Farrell, L. A., Sarzynski, M. A., Bouchard, C., Robbins, J. M., & Gerszten, R. E. (2023). Plasma Proteomic Kinetics in Response to Acute Exercise. Molecular & Cellular Proteomics, 22(8), 100601. https://doi.org/https://doi.org/10.1016/j.mcpro.2023.100601

2. Kodali, M., Madhu, L. N., Reger, R. L., Milutinovic, B., Upadhya, R., Gonzalez, J. J., Attaluri, S., Shuai, B., Gitai, D. L. G., Rao, S., Choi, J. M., Jung, S. Y., & Shetty, A. K. (2023). Intranasally administered human MSC-derived extracellular vesicles inhibit NLRP3-p38/MAPK signaling after TBI and prevent chronic brain dysfunction. Brain, Behavior, and Immunity, 108, 118–134. https://doi.org/https://doi.org/10.1016/j.bbi.2022.11.014

3. Jong Huat, T., Onraet, T., Camats-Perna, J., Newcombe, E. A., Ngo, K. C., Sue, A. N., Mirzaei, M., LaFerla, F. M., & Medeiros, R. (2023). Deletion of MyD88 in astrocytes prevents β-amyloid-induced neuropathology in mice. Glia, 71(2), 431–449. https://doi.org/10.1002/glia.24285

4. Lian, X., Liu, B., Wang, C., Wang, S., Zhuang, Y., & Li, X. (2023). Assessing of programmed cell death gene signature for predicting ovarian cancer prognosis and treatment response. Frontiers in Endocrinology, 14(June), 1–17. https://doi.org/10.3389/fendo.2023.1182776

5. Xing, H., Jiang, X., Yang, C., Tan, B., Hu, J., & Zhang, M. (2023). High expression of RPL27A predicts poor prognosis in patients with hepatocellular carcinoma. World Journal of Surgical Oncology, 21(1), 1–14. https://doi.org/10.1186/s12957-023-03102-w

6. Le, H., Dimitrakopoulou, K., Patel, H., Curtis, C., Cordero-Grande, L., Edwards, A. D., Hajnal, J., Tournier, J.-D., Deprez, M., & Cullen, H. (2023). Effect of schizophrenia common variants on infant brain volumes: cross-sectional study in 207 term neonates in developing Human Connectome Project. Translational Psychiatry, 13(1), 121. https://doi.org/10.1038/s41398-023-02413-6

7. Ma, Q., Ma, J., Cui, J., Zhang, C., Li, Y., Liu, J., Xie, K., Luo, E., Tang, C., & Zhai, M. (2023). Oxygen enrichment protects against intestinal damage and gut microbiota disturbance in rats exposed to acute high-altitude hypoxia. Frontiers in Microbiology, 14(October). https://doi.org/10.3389/fmicb.2023.1268701

8. Revol, R. S., Koistinen, N. A., Menon, P. K., Chicote-Gonzàlez, A., Iverfeldt, K., & Ström, A.-L. (2023). Alpha-secretase dependent nuclear localization of the amyloid-β precursor protein-binding protein Fe65 promotes DNA repair. Molecular and Cellular Neuroscience, 127, 103903. https://doi.org/https://doi.org/10.1016/j.mcn.2023.103903

9. Zhao, Q., Wang, N., Li, Y., Wu, Q., & Wu, L. (2023). Lnc-TMEM132D-AS1 overexpression reduces sensitivity of non-small cell lung cancer cells to osimertinib. Nan Fang Yi Ke Da Xue Xue Bao / Journal of Southern Medical University, 43(2), 242–250. https://doi.org/10.12122/j.issn.1673-4254.2023.02.12

10. Zhao, Y., Zhou, C., Wei, Y., Zhang, S., Mishra, J. S., Li, H., Lei, W., Wang, K., Kumar, S., & Zheng, J. (2023). An Endogenous Aryl Hydrocarbon Receptor Ligand Induces Preeclampsia-like Phenotypes: Transcriptome, Phosphoproteome, and Cell Functions. BioRxiv, 2023.12.20.572271. https://doi.org/10.1101/2023.12.20.572271

11. Chen, W., Li, Z., Zhong, R., Sun, W., & Chu, M. (2024). Expression profiles of oviductal mRNAs and lncRNAs in the follicular phase and luteal phase of sheep (Ovis aries) with 2 fecundity gene (FecB) genotypes. G3 Genes|Genomes|Genetics, 14(1), jkad270. https://doi.org/10.1093/g3journal/jkad270

12. Zhuang, H., Wang, X., Guo, M., Meng, Q., Liu, N., Wei, M., Shi, Y., & Deng, H. (2023). Identification of Chemokines-Related miRNAs as Potential Biomarkers in Psoriasis Based on Integrated Bioinformatics Analysis. Combinatorial Chemistry & High Throughput Screening, 26(7), 1400–1413. https://doi.org/https://doi.org/10.2174/1386207325666220819194249

13. Zhao, C., Penttinen, P., Zhang, L., Dong, L., Zhang, F., Li, Z., & Zhang, X. (2023). Mechanism of Inhibiting the Growth and Aflatoxin B1 Biosynthesis of Aspergillus flavus by Phenyllactic Acid. In Toxins (Vol. 15, Issue 6). https://doi.org/10.3390/toxins15060370

14. Ruan, G. Y., Ye, L. X., Lin, J. S., Lin, H. Y., Yu, L. R., Wang, C. Y., Mao, X. D., Zhang, S. H., & Sun, P. M. (2023). An integrated approach of network pharmacology, molecular docking, and experimental verification uncovers kaempferol as the effective modulator of HSD17B1 for treatment of endometrial cancer. Journal of Translational Medicine, 21(1), 1–19. https://doi.org/10.1186/s12967-023-04048-z

15. Fahmy, H. A., El-Shamy, S., & Farag, M. A. (2023). Comparative GC–MS based nutrients profiling of less explored legume seeds of Melilotus, Medicago, Trifolium, and Ononis analysed using chemometric tools. Scientific Reports, 13(1), 18221. https://doi.org/10.1038/s41598-023-45453-0

16. Zhao, B., Yu, H., Liu, D., Wang, J., Feng, X., He, F., Qi, T., Du, C., Wang, L., Wang, H., & Li, F. (2023). Combined Transcriptome and Metabolome Analysis Reveals Adaptive Defense Responses to DON Induction in Potato. International Journal of Molecular Sciences, 24(9). https://doi.org/10.3390/ijms24098054

17. Huang, L., Drouin, N., Causon, J., Wegrzyn, A., Castro-Perez, J., Fleming, R., Harms, A., & Hankemeier, T. (2023). Reconstruction of Glutathione Metabolism in the Neuronal Model of Rotenone-Induced Neurodegeneration Using Mass Isotopologue Analysis with Hydrophilic Interaction Liquid Chromatography-Zeno High-Resolution Multiple Reaction Monitoring. Analytical Chemistry, 95(6), 3255–3266. https://doi.org/10.1021/acs.analchem.2c04231

18. Pan, C., Yao, L., Yu, L., Qiao, Z., Tang, M., Wei, F., Huang, X., & Zhou, Y. (2023). Transcriptome and proteome analyses reveal the potential mechanism of seed dormancy release in Amomum tsaoko during warm stratification. BMC Genomics, 24(1), 99. https://doi.org/10.1186/s12864-023-09202-x

19. Huang, D., Zhong, G., Zhang, S., Jiang, K., Wang, C., Wu, J., & Wang, B. (2023). Trichome-Specific Analysis and Weighted Gene Co-Expression Correlation Network Analysis (WGCNA) Reveal Potential Regulation Mechanism of Artemisinin Biosynthesis in Artemisia annua. International Journal of Molecular Sciences, 24(10). https://doi.org/10.3390/ijms24108473

20. Esmaeilzadeh-Salestani, K., Tohidfar, M., Ghanbari Moheb Seraj, R., Khaleghdoust, B., Keres, I., Marawne, H., & Loit, E. (2023). Transcriptome profiling of barley in response to mineral and organic fertilizers. BMC Plant Biology, 23(1), 261. https://doi.org/10.1186/s12870-023-04263-2

33 Randomly Selected Articles from pubmedR Search

1. Kong, T., Fan, X., & Tran, N. T. (2023). Changes in Hemolymph Microbiota of Chinese Mitten Crab (Eriocheir sinensis) in Response to Aeromonas hydrophila or Staphylococcus aureus Infection. In Animals (Vol. 13, Issue 19). https://doi.org/10.3390/ani13193058

2. Yi, S., Feng, Y., Wang, Y., & Ma, F. (2023). Sialylation: fate decision of mammalian sperm development, fertilization, and male fertility†. Biology of Reproduction, 109(2), 137–155. https://doi.org/10.1093/biolre/ioad067

3. Jia, Z., Li, Y., Zhou, B., Xia, Q., Wang, P., Wang, X., Sun, Z., & Guo, Y. (2023). Transcriptomic profiling of human granulosa cells between women with advanced maternal age with different ovarian reserve. Journal of Assisted Reproduction and Genetics, 40(10), 2427–2437. https://doi.org/10.1007/s10815-023-02915-8

4. Kohnke, B., Kutzner, C., & Grubmüller, H. (2020). A GPU-Accelerated Fast Multipole Method for GROMACS: Performance and Accuracy. Journal of Chemical Theory and Computation, 16(11), 6938–6949. https://doi.org/10.1021/acs.jctc.0c00744

5. Barnett, M. M., Reay, W. R., Geaghan, M. P., Kiltschewskij, D. J., Green, M. J., Weidenhofer, J., Glatt, S. J., & Cairns, M. J. (2024). miRNA cargo in circulating vesicles from neurons is altered in individuals with schizophrenia and associated with severe disease. Science Advances, 9(48), eadi4386. https://doi.org/10.1126/sciadv.adi4386

6. Ayers, M., Kosar, K., Xue, Y., Goel, C., Carson, M., Lee, E., Liu, S., Brooks, E., Cornuet, P., Oertel, M., Bhushan, B., & Nejak-Bowen, K. (2023). Inhibiting Wnt Signaling Reduces Cholestatic Injury by Disrupting the Inflammatory Axis. Cmgh, 16(6), 895–921. https://doi.org/10.1016/j.jcmgh.2023.08.004

7. Kang, K., Wu, J., Mao, Y., Kai, J., Chen, S., & Xiong, F. (2023). Role of SLC44A3-AS1 Enhancer RNA in Esophageal Cancer Prognosis. Journal of the College of Physicians and Surgeons Pakistan, 33(9), 964–971. https://doi.org/10.29271/jcpsp.2023.09.964

8. Zheng, C., Nie, H., Pan, M., Fan, W., Pi, D., Liang, Z., Liu, D., Wang, F., Yang, Q., & Zhang, Y. (2024). Chaihu Shugan powder influences nonalcoholic fatty liver

disease in rats in remodeling microRNAome and decreasing fatty acid synthesis. Journal of Ethnopharmacology, 318, 116967. https://doi.org/https://doi.org/10.1016/j.jep.2023.116967

9. Liu, J., Liu, J., Zhang, P., Wang, Q., Li, L., Xie, H., Li, H., Wang, H., Cheng, S., & Qin, P. (2023). Elucidating the Differentiation Synthesis Mechanisms of Differently Colored Resistance Quinoa Seedings Using Metabolite Profiling and Transcriptome Analysis. Metabolites, 13(10). https://doi.org/10.3390/metabo13101065

10. Chen, K., Shi, Y., & Zhu, H. (2023). Analysis of the role of glucose metabolism-related genes in dilated cardiomyopathy based on bioinformatics. Journal of Thoracic Disease; Vol 15, No 7 (July 31, 2023): Journal of Thoracic Disease. https://jtd.amegroups.org/article/view/77277

11. Yin, M., Zhao, S., Lai, J., Yang, X., Dong, B., Zhu, Y., & Zhang, Y. (2023). Oxygen-insensitive nitroreductase bacteria-mediated degradation of TNT and proteomic analysis. Environmental Science and Pollution Research, 30(54), 116227–116238. https://doi.org/10.1007/s11356-023-30568-8

12. Dai, W., Zheng, P., Wu, J., Chen, S., Deng, M., Tong, X., Liu, F., Shang, X., & Qian, K. (2024). Integrated analysis of single-cell RNA-seq and chipset data unravels PANoptosis-related genes in sepsis. Frontiers in Immunology, 14. https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2023.1247131

13. Fu, R., Liu, H., Zhang, Y., Mao, L., Zhu, L., Jiang, H., Zhang, L., & Liu, X. (2024). Imidacloprid affects the visual behavior of adult zebrafish (Danio rerio) by mediating the expression of opsin and phototransduction genes and altering the metabolism of neurotransmitters. Science of The Total Environment, 910, 168572. https://doi.org/https://doi.org/10.1016/j.scitotenv.2023.168572

14. Ma, Y., Huang, X., Du, H., Yang, J., Guo, F., & Wu, F. (2024). Impacts, causes and biofortification strategy of rice selenium deficiency based on publication collection. Science of The Total Environment, 912, 169619. https://doi.org/https://doi.org/10.1016/j.scitotenv.2023.169619

15. Sales, L. P., Hounkpe, B. W., Perez, M. O., Caparbo, V. F., Domiciano, D. S., Borba, E. F., Schett, G., Figueiredo, C. P., & Pereira, R. M. R. (2023). Transcriptomic characterization of classical monocytes highlights the involvement of immuno-inflammation in bone erosion in Rheumatoid Arthritis. Frontiers in Immunology, 14(October), 1–15. https://doi.org/10.3389/fimmu.2023.1251034

16. Kiewisz, R., Baum, D., Müller-Reichert, T., & Fabig, G. (2023). Serial-section Electron Tomography and Quantitative Analysis of Microtubule Organization in 3D-reconstructed Mitotic Spindles. Bio-Protocol, 13(20), e4849. https://doi.org/10.21769/BioProtoc.4849

17. Li, S. X., Yang, Y. J., & Chen, D. L. (2023). Structural Evolution and Electronic Properties of Two Sulfur Atom-Doped Boron Clusters. ACS Omega, 8(33), 30757–30767. https://doi.org/10.1021/acsomega.3c04967

18. Thorel, M., Obregon, D., Mulot, B., Maitre, A., Mateos-Hernandez, L., Moalic, P.-Y., Wu-Chuang, A., Cabezas-Cruz, A., & Leclerc, A. (2023). Conserved core microbiota in managed and free-ranging Loxodonta africana elephants. Frontiers in Microbiology, 14. https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2023.1247719

19. Gao, X., An, J., Yu, C., Zha, X., & Tian, Y. (2023). Dietary sources apportionment and health risk assessment for trace elements among residents of the Tethys-Himalayan tectonic domain in Tibet, China. Environmental Geochemistry and Health, 45(11), 8015–8030. https://doi.org/10.1007/s10653-023-01706-5

20. Barraza, F., Javed, M. B., Noernberg, T., Schultz, J., & Shotyk, W. (2024). Spatial variation and chemical reactivity of dusts from open-pit bitumen mining using trace elements in snow. Chemosphere, 350, 141081. https://doi.org/https://doi.org/10.1016/j.chemosphere.2023.141081

21. Xu, Y., Xu, J., Chen, S., Zhou, A., Huang, G., Huang, S., Yu, D., & Wu, B. (2023). Identifying potential pathogenesis and immune infiltration in diabetic foot ulcers using bioinformatics and in vitro analyses. BMC Medical Genomics, 16(1), 1–14. https://doi.org/10.1186/s12920-023-01741-2

22. Zhang, W., Li, G. S., Gan, X. Y., Huang, Z. G., He, R. Q., Huang, H., Li, D. M., Tang, Y. L., Tang, D., Zou, W., Liu, J., Dang, Y. W., Chen, G., Zhou, H. F., Kong, J. L., & Lu, H. P. (2023). MMP12 serves as an immune cell–related marker of disease status and prognosis in lung squamous cell carcinoma. PeerJ, 11, 1–24. https://doi.org/10.7717/peerj.15598

23. Vrinda, P. K., Amal, R., Abhirami, N., Mini, D. A., Kumar, V. J. R., & Devipriya, S. P. (2023). Co-exposure of microplastics and heavy metals in the marine environment and remediation techniques: a comprehensive review. Environmental Science and Pollution Research, 30(54), 114822–114843. https://doi.org/10.1007/s11356-023-30679-2

24. Liu, M., Hu, F., Liu, L., Lu, X., Li, R., Wang, J., Wu, J., Ma, L., Pu, Y., Fang, Y., Yang, G., Wang, W., & Sun, W. (2023). Physiological Analysis and Genetic Mapping of Short Hypocotyl Trait in Brassica napus L. International Journal of Molecular Sciences, 24(20). https://doi.org/10.3390/ijms242015409

25. Wang, R., & Yang, Y. M. (2023). Identification of potential biomarkers for idiopathic pulmonary fibrosis and validation of TDO2 as a potential therapeutic target. World Journal of Cardiology, 15(6), 293–308. https://doi.org/10.4330/wjc.v15.i6.293

26. Linkova, N., Khavinson, V., Diatlova, A., Petukhov, M., Vladimirova, E., Sukhareva, M., & Ilina, A. (2023). The Influence of KE and EW Dipeptides in the Composition of the Thymalin Drug on Gene Expression and Protein Synthesis Involved in the Pathogenesis of COVID-19. International Journal of Molecular Sciences, 24(17). https://doi.org/10.3390/ijms241713377

27. Wang, S., Zeng, J., Li, P., Wang, C., Zhou, A., Gao, L., Kong, X., Li, X., Yue, X., & Luo, J. (2023). Distribution characteristics, risk assessment, and relevance

with surrounding soil of heavy metals in coking solid wastes from coking plants in Shanxi, China. Environmental Monitoring and Assessment, 195(12), 1399. https://doi.org/10.1007/s10661-023-11938-8

28. He, J., Jia, W., Lin, Z., Zhang, Y., Zhao, Y., & Fang, Y. (2023). Improving the quality and processing efficiency of beef jerky via drying in confined conditions of pre-stretching. Food Research International, 172, 113171. https://doi.org/https://doi.org/10.1016/j.foodres.2023.113171

29. Lee, J. Y., Harney, D. J., Teo, J. D., Kwok, J. B., Sutherland, G. T., Larance, M., & Don, A. S. (2023). The major TMEM106B dementia risk allele affects TMEM106B protein levels, fibril formation, and myelin lipid homeostasis in the ageing human hippocampus. Molecular Neurodegeneration, 18(1), 63. https://doi.org/10.1186/s13024-023-00650-3

30. Yuan, D., Huang, B., Gu, M., Qin, B., Su, Z., Dai, K., Peng, F., & Jiang, Y. (2023). Exploring Shared Genetic Signatures of Alzheimer's Disease and Multiple Sclerosis: A Bioinformatic Analysis Study. European Neurology, 86(6), 363–376. https://doi.org/10.1159/000533397

31. Sakr, M. A. S., Saad, M. A., Abdelsalam, H., Teleb, N. H., & Zhang, Q. (2023). Electronic and optical properties of chemically modified 2D GaAs nanoribbons. Scientific Reports, 13(1), 15535. https://doi.org/10.1038/s41598-023-42855-y

32. Zhang, X., Zhao, P., Ma, M., Wu, H., Liu, R., Liu, Z., Cai, Z., Liu, M., Xie, F., & Ma, X. (2023). Missing link between tissue specific expressing pattern of ERβ and the clinical manifestations in LGBLEL. Frontiers in Medicine, 10. https://www.frontiersin.org/journals/medicine/articles/10.3389/fmed.2023.1168977

33. Park, A. Y., Han, M.-R., Seo, B. K., Ju, H.-Y., Son, G. S., Lee, H. Y., Chang, Y. W., Choi, J., Cho, K. R., Song, S. E., Woo, O. H., & Park, H. S. (2023). MRI-based breast cancer radiogenomics using RNA profiling: association with subtypes in a single-center prospective study. Breast Cancer Research, 25(1), 79. https://doi.org/10.1186/s13058-023-01668-7

44 articles from Focused Search in *Cell*, *Nature* and *Science*

1. Walker, J. T., Saunders, D. C., Rai, V., Chen, H.-H., Orchard, P., Dai, C., Pettway, Y. D., Hopkirk, A. L., Reihsmann, C. V, Tao, Y., Fan, S., Shrestha, S., Varshney, A., Petty, L. E., Wright, J. J., Ventresca, C., Agarwala, S., Aramandla, R., Poffenberger, G., ... Consortium, T. H. (2023). Genetic risk converges on regulatory networks mediating early type 2 diabetes. Nature, 624(7992), 621–629. https://doi.org/10.1038/s41586-023-06693-2

2. Sun, B. B., Chiou, J., Traylor, M., Benner, C., Hsu, Y.-H., Richardson, T. G., Surendran, P., Mahajan, A., Robins, C., Vasquez-Grinnell, S. G., Hou, L., Kvikstad, E. M., Burren, O. S., Davitte, J., Ferber, K. L., Gillies, C. E., Hedman, Å. K., Hu, S., Lin, T., ... Center, R. G. (2023). Plasma proteomic associations with genetics and health in the UK Biobank. Nature, 622(7982), 329–338. https://doi.org/10.1038/s41586-023-06592-6

3. Allen, W. E., Blosser, T. R., Sullivan, Z. A., Dulac, C., & Zhuang, X. (2023). Molecular and spatial signatures of mouse brain aging at single-cell resolution. Cell, 186(1), 194-208.e18. https://doi.org/10.1016/j.cell.2022.12.010

4. Zu, S., Li, Y. E., Wang, K., Armand, E. J., Mamde, S., Amaral, M. L., Wang, Y., Chu, A., Xie, Y., Miller, M., Xu, J., Wang, Z., Zhang, K., Jia, B., Hou, X., Lin, L., Yang, Q., Lee, S., Li, B., ... Ren, B. (2023). Single-cell analysis of chromatin accessibility in the adult mouse brain. Nature, 624(7991), 378–389. https://doi.org/10.1038/s41586-023-06824-9

5. Shih, J., Sarmashghi, S., Zhakula-Kostadinova, N., Zhang, S., Georgis, Y., Hoyt, S. H., Cuoco, M. S., Gao, G. F., Spurr, L. F., Berger, A. C., Ha, G., Rendo, V., Shen, H., Meyerson, M., Cherniack, A. D., Taylor, A. M., & Beroukhim, R. (2023). Cancer aneuploidies are shaped primarily by effects on tumour fitness. Nature, 619(7971), 793–800. https://doi.org/10.1038/s41586-023-06266-3

6. Tsai, S.-M., Lee, E. K. H., Powell, D., Gao, P., Zhang, X., Moses, J., Hébrard, E., Venot, O., Parmentier, V., Jordan, S., Hu, R., Alam, M. K., Alderson, L., Batalha, N. M., Bean, J. L., Benneke, B., Bierson, C. J., Brady, R. P., Carone, L., ... Yurchenko, S. N. (2023). Photochemically produced SO2 in the atmosphere of WASP-39b. Nature, 617(7961), 483–487. https://doi.org/10.1038/s41586-023-05902-29

7. Dileep, V., Boix, C. A., Mathys, H., Marco, A., Welch, G. M., Meharena, H. S., Loon, A., Jeloka, R., Peng, Z., Bennett, D. A., Kellis, M., & Tsai, L. H. (2023). Neuronal DNA double-strand breaks lead to genome structural variations and 3D genome disruption in neurodegeneration. Cell, 186(20), 4404-4421.e20. https://doi.org/10.1016/j.cell.2023.08.038

8. Terekhanova, N. V, Karpova, A., Liang, W.-W., Strzalkowski, A., Chen, S., Li, Y., Southard-Smith, A. N., Iglesia, M. D., Wendl, M. C., Jayasinghe, R. G., Liu, J., Song, Y., Cao, S., Houston, A., Liu, X., Wyczalkowski, M. A., Lu, R. J.-H., Caravan, W., Shinkle, A., ... Ding, L. (2023). Epigenetic regulation during cancer transitions across 11 tumour types. Nature, 623(7986), 432–441. https://doi.org/10.1038/s41586-023-06682-5

9. Rapin, W., Dromart, G., Clark, B. C., Schieber, J., Kite, E. S., Kah, L. C., Thompson, L. M., Gasnault, O., Lasue, J., Meslin, P.-Y., Gasda, P. J., & Lanza, N. L. (2023). Sustained wet–dry cycling on early Mars. Nature, 620(7973), 299–302. https://doi.org/10.1038/s41586-023-06220-3

10. Chen, A., Sun, Y., Lei, Y., Li, C., Liao, S., Meng, J., Bai, Y., Liu, Z., Liang, Z., Zhu, Z., Yuan, N., Yang, H., Wu, Z., Lin, F., Wang, K., Li, M., Zhang, S., Yang, M., Fei, T., ... Li, C. (2023). Single-cell spatial transcriptome reveals cell-type organization in the macaque cortex. Cell, 186(17), 3726-3743.e24. https://doi.org/10.1016/j.cell.2023.06.009

11. Jbara, A., Lin, K.-T., Stossel, C., Siegfried, Z., Shqerat, H., Amar-Schwartz, A., Elyada, E., Mogilevsky, M., Raitses-Gurevich, M., Johnson, J. L., Yaron, T. M., Ovadia, O., Jang, G. H., Danan-Gotthold, M., Cantley, L. C., Levanon, E. Y.,

Gallinger, S., Krainer, A. R., Golan, T., & Karni, R. (2023). RBFOX2 modulates a metastatic signature of alternative splicing in pancreatic cancer. Nature, 617(7959), 147–153. https://doi.org/10.1038/s41586-023-05820-3

12. Caglayan, E., Ayhan, F., Liu, Y., Vollmer, R. M., Oh, E., Sherwood, C. C., Preuss, T. M., Yi, S. V, & Konopka, G. (2023). Molecular features driving cellular complexity of human brain evolution. Nature, 620(7972), 145–153. https://doi.org/10.1038/s41586-023-06338-4

13. Erwin, G. S., Gürsoy, G., Al-Abri, R., Suriyaprakash, A., Dolzhenko, E., Zhu, K., Hoerner, C. R., White, S. M., Ramirez, L., Vadlakonda, A., Vadlakonda, A., von Kraut, K., Park, J., Brannon, C. M., Sumano, D. A., Kirtikar, R. A., Erwin, A. A., Metzner, T. J., Yuen, R. K. C., … Snyder, M. P. (2023). Recurrent repeat expansions in human cancer genomes. Nature, 613(7942), 96–102. https://doi.org/10.1038/s41586-022-05515-1

14. Loyfer, N., Magenheim, J., Peretz, A., Cann, G., Bredno, J., Klochendler, A., Fox-Fisher, I., Shabi-Porat, S., Hecht, M., Pelet, T., Moss, J., Drawshy, Z., Amini, H., Moradi, P., Nagaraju, S., Bauman, D., Shveiky, D., Porat, S., Dior, U., … Kaplan, T. (2023). A DNA methylation atlas of normal human cell types. Nature, 613(7943), 355–364. https://doi.org/10.1038/s41586-022-05580-6

15. Zhou, P., Shi, H., Huang, H., Sun, X., Yuan, S., Chapman, N. M., Connelly, J. P., Lim, S. A., Saravia, J., KC, A., Pruett-Miller, S. M., & Chi, H. (2023). Single-cell CRISPR screens in vivo map T cell fate regulomes in cancer. Nature, 624(7990), 154–163. https://doi.org/10.1038/s41586-023-06733-x

16. Servellita, V., Sotomayor Gonzalez, A., Lamson, D. M., Foresythe, A., Huh, H. J., Bazinet, A. L., Bergman, N. H., Bull, R. L., Garcia, K. Y., Goodrich, J. S., Lovett, S. P., Parker, K., Radune, D., Hatada, A., Pan, C.-Y., Rizzo, K., Bertumen, J. B., Morales, C., Oluniyi, P. E., … Group, P. H. of U. E. W. (2023). Adeno-associated virus type 2 in US children with acute severe hepatitis. Nature, 617(7961), 574–580. https://doi.org/10.1038/s41586-023-05949-1

17. Nam, C. H., Youk, J., Kim, J. Y., Lim, J., Park, J. W., Oh, S. A., Lee, H. J., Park, J. W., Won, H., Lee, Y., Jeong, S.-Y., Lee, D.-S., Oh, J. W., Han, J., Lee, J., Kwon, H. W., Kim, M. J., & Ju, Y. S. (2023). Widespread somatic L1 retrotransposition in normal colorectal epithelium. Nature, 617(7961), 540–547. https://doi.org/10.1038/s41586-023-06046-z

18. Shalon, D., Culver, R. N., Grembi, J. A., Folz, J., Treit, P. V, Shi, H., Rosenberger, F. A., Dethlefsen, L., Meng, X., Yaffe, E., Aranda-Díaz, A., Geyer, P. E., Mueller-Reif, J. B., Spencer, S., Patterson, A. D., Triadafilopoulos, G., Holmes, S. P., Mann, M., Fiehn, O., … Huang, K. C. (2023). Profiling the human intestinal environment under physiological conditions. Nature, 617(7961), 581–591. https://doi.org/10.1038/s41586-023-05989-7

19. Setton, J., Hadi, K., Choo, Z.-N., Kuchin, K. S., Tian, H., Da Cruz Paula, A., Rosiene, J., Selenica, P., Behr, J., Yao, X., Deshpande, A., Sigouros, M., Manohar, J., Nauseef, J. T., Mosquera, J.-M., Elemento, O., Weigelt, B., Riaz, N., Reis-Filho, J. S., … Imieliński, M. (2023). Long-molecule scars of backup DNA repair

in BRCA1- and BRCA2-deficient cancers. Nature, 621(7977), 129–137. https://doi.org/10.1038/s41586-023-06461-2

20. Gurtner, A., Borrelli, C., Gonzalez-Perez, I., Bach, K., Acar, I. E., Núñez, N. G., Crepaz, D., Handler, K., Vu, V. P., Lafzi, A., Stirm, K., Raju, D., Gschwend, J., Basler, K., Schneider, C., Slack, E., Valenta, T., Becher, B., Krebs, P., ... Arnold, I. C. (2023). Active eosinophils regulate host defence and immune responses in colitis. Nature, 615(7950), 151–157. https://doi.org/10.1038/s41586-022-05628-7

21. Sneppen, A., Watson, D., Bauswein, A., Just, O., Kotak, R., Nakar, E., Poznanski, D., & Sim, S. (2023). Spherical symmetry in the kilonova AT2017gfo/GW170817. Nature, 614(7948), 436–439. https://doi.org/10.1038/s41586-022-05616-x

22. Morris, J. A., Caragine, C., Daniloski, Z., Domingo, J., Barry, T., Lu, L., Davis, K., Ziosi, M., Glinos, D. A., Hao, S., Mimitou, E. P., Smibert, P., Roeder, K., Katsevich, E., Lappalainen, T., & Sanjana, N. E. (2024). Discovery of target genes and pathways at GWAS loci by pooled single-cell CRISPR screens. Science, 380(6646), eadh7699. https://doi.org/10.1126/science.adh7699

23. Al Bakir, M., Huebner, A., Martínez-Ruiz, C., Grigoriadis, K., Watkins, T. B. K., Pich, O., Moore, D. A., Veeriah, S., Ward, S., Laycock, J., Johnson, D., Rowan, A., Razaq, M., Akther, M., Naceur-Lombardelli, C., Prymas, P., Toncheva, A., Hessey, S., Dietzen, M., ... Consortium, Tracer. (2023). The evolution of non-small cell lung cancer metastases in TRACERx. Nature, 616(7957), 534–542. https://doi.org/10.1038/s41586-023-05729-x

24. van Oostrum, M., Blok, T. M., Giandomenico, S. L., tom Dieck, S., Tushev, G., Fürst, N., Langer, J. D., & Schuman, E. M. (2023). The proteomic landscape of synaptic diversity across brain regions and cell types. Cell, 186(24), 5411-5427.e23. https://doi.org/10.1016/j.cell.2023.09.028

25. Velmeshev, D., Perez, Y., Yan, Z., Valencia, J. E., Castaneda-Castellanos, D. R., Wang, L., Schirmer, L., Mayer, S., Wick, B., Wang, S., Nowakowski, T. J., Paredes, M., Huang, E. J., & Kriegstein, A. R. (2024). Single-cell analysis of prenatal and postnatal human cortical development. Science, 382(6667), eadf0834. https://doi.org/10.1126/science.adf0834

26. Gao, Y., Yang, X., Chen, H., Tan, X., Yang, Z., Deng, L., Wang, B., Kong, S., Li, S., Cui, Y., Lei, C., Wang, Y., Pan, Y., Ma, S., Sun, H., Zhao, X., Shi, Y., Yang, Z., Wu, D., ... (CPC), C. P. C. (2023). A pangenome reference of 36 Chinese populations. Nature, 619(7968), 112–121. https://doi.org/10.1038/s41586-023-06173-7

27. Kun, E., Javan, E. M., Smith, O., Gulamali, F., de la Fuente, J., Flynn, B. I., Vajrala, K., Trutner, Z., Jayakumar, P., Tucker-Drob, E. M., Sohail, M., Singh, T., & Narasimhan, V. M. (2024). The genetic architecture and evolution of the human skeletal form. Science, 381(6655), eadf8009. https://doi.org/10.1126/science.adf8009

28. Tyshkovskiy, A., Ma, S., Shindyapina, A. V, Tikhonov, S., Lee, S.-G., Bozaykut, P., Castro, J. P., Seluanov, A., Schork, N. J., Gorbunova, V., Dmitriev, S. E., Miller, R. A., & Gladyshev, V. N. (2023). Distinct longevity mechanisms across and within species and their association with aging. Cell, 186(13), 2929-2949.e20. https://doi.org/10.1016/j.cell.2023.05.002

29. Li, Z., Chen, S., Zhao, L., Huang, G., Xu, H., Yang, X., Wang, P., Gao, N., & Sui, S.-F. (2023). Nuclear export of pre-60S particles through the nuclear pore complex. Nature, 618(7964), 411–418. https://doi.org/10.1038/s41586-023-06128-y

30. Paredes, A., Justo-Méndez, R., Jiménez-Blasco, D., Núñez, V., Calero, I., Villalba-Orero, M., Alegre-Martí, A., Fischer, T., Gradillas, A., Sant'Anna, V. A. R., Were, F., Huang, Z., Hernansanz-Agustín, P., Contreras, C., Martínez, F., Camafeita, E., Vázquez, J., Ruiz-Cabello, J., Area-Gómez, E., ... Ricote, M. (2023). γ-Linolenic acid in maternal milk drives cardiac metabolic maturation. Nature, 618(7964), 365–373. https://doi.org/10.1038/s41586-023-06068-7

31. Cui, A., Huang, T., Li, S., Ma, A., Pérez, J. L., Sander, C., Keskin, D. B., Wu, C. J., Fraenkel, E., & Hacohen, N. (2024). Dictionary of immune responses to cytokines at single-cell resolution. Nature, 625(7994), 377–384. https://doi.org/10.1038/s41586-023-06816-9

32. Tan, L., Shi, J., Moghadami, S., Parasar, B., Wright, C. P., Seo, Y., Vallejo, K., Cobos, I., Duncan, L., Chen, R., & Deisseroth, K. (2023). Lifelong restructuring of 3D genome architecture in cerebellar granule cells. Science, 381(6662), 1112–1119. https://doi.org/10.1126/science.adh3253

33. Chang, C.-Y., Bajić, D., Vila, J. C. C., Estrela, S., & Sanchez, A. (2023). Emergent coexistence in multispecies microbial communities. Science, 381(6655), 343–348. https://doi.org/10.1126/science.adg0727

34. Langlieb, J., Sachdev, N. S., Balderrama, K. S., Nadaf, N. M., Raj, M., Murray, E., Webber, J. T., Vanderburg, C., Gazestani, V., Tward, D., Mezias, C., Li, X., Flowers, K., Cable, D. M., Norton, T., Mitra, P., Chen, F., & Macosko, E. Z. (2023). The molecular cytoarchitecture of the adult mouse brain. Nature, 624(7991), 333–342. https://doi.org/10.1038/s41586-023-06818-7

35. Kurki, M. I., Karjalainen, J., Palta, P., Sipilä, T. P., Kristiansson, K., Donner, K. M., Reeve, M. P., Laivuori, H., Aavikko, M., Kaunisto, M. A., Loukola, A., Lahtela, E., Mattsson, H., Laiho, P., Della Briotta Parolo, P., Lehisto, A. A., Kanai, M., Mars, N., Rämö, J., ... FinnGen. (2023). FinnGen provides genetic insights from a well-phenotyped isolated population. Nature, 613(7944), 508–518. https://doi.org/10.1038/s41586-022-05473-8

36. Zhou, J., Zhang, Z., Wu, M., Liu, H., Pang, Y., Bartlett, A., Peng, Z., Ding, W., Rivkin, A., Lagos, W. N., Williams, E., Lee, C.-T., Miyazaki, P. A., Aldridge, A., Zeng, Q., Salinda, J. L. A., Claffey, N., Liem, M., Fitzpatrick, C., ... Callaway, E. M. (2023). Brain-wide correspondence of neuronal epigenomics and distant projections. Nature, 624(7991), 355–365. https://doi.org/10.1038/s41586-023-06823-w

37. Michaelis, A. C., Brunner, A.-D., Zwiebel, M., Meier, F., Strauss, M. T., Bludau, I., & Mann, M. (2023). The social and structural architecture of the yeast protein interactome. Nature, 624(7990), 192–200. https://doi.org/10.1038/s41586-023-06739-5

38. Kim, C. N., Shin, D., Wang, A., & Nowakowski, T. J. (2024). Spatiotemporal molecular dynamics of the developing human thalamus. Science, 382(6667), eadf9941. https://doi.org/10.1126/science.adf9941

39. Li, Z., Zhang, Y. G., Torres, M., & Mills, B. J. W. (2023). Neogene burial of organic carbon in the global ocean. Nature, 613(7942), 90–95. https://doi.org/10.1038/s41586-022-05413-6

40. Chen, S., Francioli, L. C., Goodrich, J. K., Collins, R. L., Kanai, M., Wang, Q., Alföldi, J., Watts, N. A., Vittal, C., Gauthier, L. D., Poterba, T., Wilson, M. W., Tarasova, Y., Phu, W., Grant, R., Yohannes, M. T., Koenig, Z., Farjoun, Y., Banks, E., … Consortium, G. A. D. (2024). A genomic mutational constraint map using variation in 76,156 human genomes. Nature, 625(7993), 92–100. https://doi.org/10.1038/s41586-023-06045-0

41. Li, C., Fleck, J. S., Martins-Costa, C., Burkard, T. R., Themann, J., Stuempflen, M., Peer, A. M., Vertesy, Á., Littleboy, J. B., Esk, C., Elling, U., Kasprian, G., Corsini, N. S., Treutlein, B., & Knoblich, J. A. (2023). Single-cell brain organoid screening identifies developmental defects in autism. Nature, 621(7978), 373–380. https://doi.org/10.1038/s41586-023-06473-y

42. Han, S., Lee, M., Shin, Y., Giovanni, R., Chakrabarty, R. P., Herrerias, M. M., Dada, L. A., Flozak, A. S., Reyfman, P. A., Khuder, B., Reczek, C. R., Gao, L., Lopéz-Barneo, J., Gottardi, C. J., Budinger, G. R. S., & Chandel, N. S. (2023). Mitochondrial integrated stress response controls lung epithelial cell fate. Nature, 620(7975), 890–897. https://doi.org/10.1038/s41586-023-06423-8

43. Tang, F., Li, J., Qi, L., Liu, D., Bo, Y., Qin, S., Miao, Y., Yu, K., Hou, W., Li, J., Peng, J., Tian, Z., Zhu, L., Peng, H., Wang, D., & Zhang, Z. (2023). A pan-cancer single-cell panorama of human natural killer cells. Cell, 186(19), 4235-4251.e20. https://doi.org/10.1016/j.cell.2023.07.034

44. Zhang, F., Jonsson, A. H., Nathan, A., Millard, N., Curtis, M., Xiao, Q., Gutierrez-Arcelus, M., Apruzzese, W., Watts, G. F. M., Weisenfeld, D., Nayar, S., Rangel-Moreno, J., Meednu, N., Marks, K. E., Mantel, I., Kang, J. B., Rumker, L., Mears, J., Slowikowski, K., … Network, A. M. P. R. (2023). Deconstruction of rheumatoid arthritis synovium defines inflammatory subtypes. Nature, 623(7987), 616–624. https://doi.org/10.1038/s41586-023-06708-y

Acknowledgements

[redacted]

Curriculum Vitae

[redacted]