

DeeDeeExperiment: building an infrastructure for integrating and managing omics data analysis results in R/Bioconductor

Najla Abassi^{1,*}, Lea Schwarz¹, Edoardo Filippi^{1,2}, Federico Marini^{1,3,*}

¹Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center Mainz, Mainz, 55131, Germany

²Department of Nephrology, Rheumatology and Kidney Transplantation, University Medical Center Mainz, Mainz, 55131, Germany

³Research Center for Immunotherapy (FZI) Mainz, Mainz, 55131, Germany

*Corresponding authors. Najla Abassi, Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center Mainz, Langenbeckstrasse 1, 55131 Mainz, Germany. E-mail: najla.abassi@uni-mainz.de; Federico Marini, Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center Mainz, Langenbeckstrasse 1, 55131 Mainz, Germany. E-mail: marinif@uni-mainz.de

Associate Editor: Russell Schwartz

Abstract

Summary: Modern omics experiments now involve multiple conditions and complex designs, producing an increasingly large set of differential expression and functional enrichment analysis results. However, no standardized data structure exists to store and contextualize these results together with their metadata, leaving researchers with an unmanageable and potentially non-reproducible collection of results that are difficult to navigate and/or share. Here we introduce DeeDeeExperiment, a new S4 class for managing and storing omics data analysis results, implemented within the Bioconductor ecosystem, which promotes interoperability, reproducibility and good documentation. This class extends the widely used SingleCellExperiment object by introducing dedicated slots for Differential Expression (DEA) and Functional Enrichment Analysis (FEA) results, allowing users to organize, store, and retrieve information on multiple contrasts and associated metadata within a single data object, ultimately streamlining the management and interpretation of many omics datasets.

Availability and implementation: DeeDeeExperiment is available on Bioconductor under the MIT license (<https://bioconductor.org/packages/DeeDeeExperiment>), with its development version also available on Github (<https://github.com/imbei-mainz/DeeDeeExperiment>).

1 Introduction

Over the past decade, omics technologies have advanced the study of human health and diseases, as well as a wide range of model organisms generating data at unprecedented scale (Van den Berge *et al.* 2019, Chen *et al.* 2023, Smirnov *et al.* 2023). As the volume of these data grows, so does the challenge of analyzing and managing the increasingly heterogeneous results they produce. In practice, many of these results take the form of tables generated from differential expression analysis (DEA) and functional enrichment analysis (FEA), such as Over-Representation Analysis and Gene Set Enrichment Analysis (Alexa *et al.* 2006, Robinson *et al.* 2010, Yu *et al.* 2012, Chen *et al.* 2013, Love *et al.* 2014, Ritchie *et al.* 2015, Korotkevich *et al.* 2021). These steps are central in most omics workflows, enabling researchers to detect and interpret biological differences between conditions. However, as the number of conditions and analysis tools increases, these outputs can quickly accumulate

in large numbers, making their management overwhelming even for expert practitioners. This challenge is evident in bulk RNA-seq and other omics modalities (e.g. proteomics or metabolomics), where complex experimental designs can generate extensive results (Hajjar *et al.* 2023, Peng *et al.* 2024). This becomes even amplified in single-cell RNA-seq studies, where the pseudobulk analysis (i.e. aggregated at the cluster-sample level) yields numerous results tables across multiple conditions/cell types (Crowell *et al.* 2020).

Despite the availability of numerous DEA and FEA frameworks implemented within the Bioconductor project, which relies on interoperable S4 classes for omics analyses, there is currently no standardized data structure for organizing, linking, and contextualizing multiple DEA and FEA results together with their associated metadata. Without such infrastructure, it becomes difficult to keep track of the analytical context, recall parameters, or share results in a reproducible manner, particularly in collaborative settings.

Here, we describe `DeeDeeExperiment`, a new `S4` class that extends the widely adopted `SingleCellExperiment` Bioconductor object (Amezquita *et al.* 2020). The class provides a standardized data structure for organizing DEA and FEA results alongside expression data and metadata. Our framework enables users to efficiently retrieve and explore outputs across multiple contrasts in a structured and reproducible manner. Being built within the Bioconductor ecosystem, `DeeDeeExperiment` leverages its long-standing principles of interoperability, continuous testing, optimization, and comprehensive documentation (Huber *et al.* 2015), ensuring compatibility with existing downstream tools and workflows, including those for interpretation and interactive exploration (Rue-Albrecht *et al.* 2018, Marini *et al.* 2021, Ludt *et al.* 2022).

2 Implementation

The `DeeDeeExperiment` package includes the class definition, along with methods to create, access, and manipulate `DeeDeeExperiment` objects. Objects can be created with the constructor function `DeeDeeExperiment()`. Since the class inherits from `SingleCellExperiment`, it retains all its core slots, including: (i) `assays`, a matrix-like slot (e.g. a matrix or `DelayedArray`) that holds experimental data such as raw counts and/or normalized values across a common set of samples or cells, (ii) `rowData`, a `DataFrame` object describing the features under inspection (e.g. gene identifiers and all the associated information), (iii) `colData`, a `DataFrame` describing the samples

or cells (such as experimental condition, batch, donor, cell type annotation, or quality control metrics), (iv) `metadata`, for additional information, and (v) the `reducedDims`, which stores reduced-dimensional representations of the data, such as PCA, t-SNE or UMAP (Maaten and Hinton 2008, Jolliffe and Cadima 2016, McInnes *et al.* 2018). In addition, `DeeDeeExperiment` introduces two new slots: (vi) `dea`, which stores Differential Expression Analysis (DEA) results as standard objects from `DESeq2` (Love *et al.* 2014), `edgeR` (Robinson *et al.* 2010), and `limma` (Ritchie *et al.* 2015) or a `data.frame` provided by the user that contains at least three required columns: `log2FoldChange`, `pvalue`, and `padj`. Additionally, the (vii) `fea` slot stores Functional Enrichment Analysis (FEA) results from commonly used packages such as `topGO` (Alexa *et al.* 2006), `clusterProfiler` (Yu *et al.* 2012), or `GeneTonic` (Marini *et al.* 2021) among others (Fig. 1). The `dea` and `fea` slots are implemented as a set of named contrasts, each carrying basic metadata such as the package and version used to generate the results. To ensure efficient storage, the full original DEA results objects for each comparison are stored in the `metadata` slot of `DeeDeeExperiment`. Each contrast entry retains an internal pointer referencing its associated original object, allowing users to retrieve it seamlessly when needed. Feature-level DEA statistics (`logFC`, `P-value`, and adjusted `P-value`) for each contrast are embedded into the `rowData` slot of `DeeDeeExperiment`.

Our implementation provides methods to add, remove, and retrieve stored analysis results for different contrasts, as well as a `summary()` method for inspecting them. The `summary`

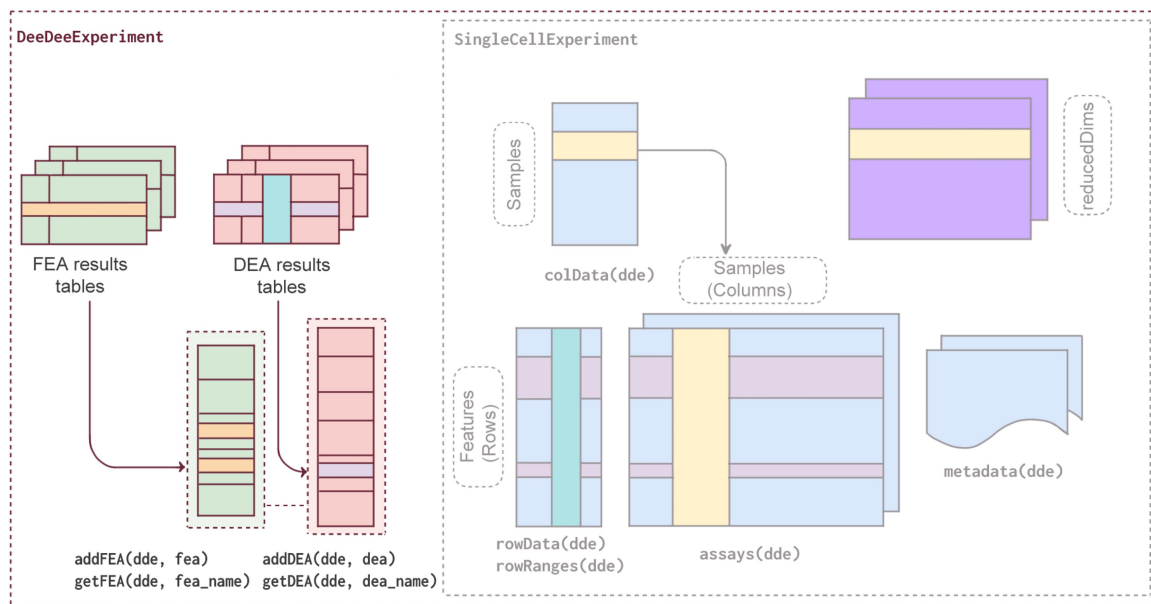


Figure 1 Anatomy of `DeeDeeExperiment` class. It extends the `SingleCellExperiment` class (right panel), inheriting its core slots (`assays`, `rowData`, `colData`, `reducedDims`, `metadata`), and adds two new slots: `dea`, for storing Differential Expression Analysis (DEA) results, and `fea`, for storing Functional Enrichment Analysis (FEA) results. Both slots store results as a list of named contrasts/comparisons (hereby used interchangeably). Their names are provided by the user (defaulting to the object name if not specified) and are enforced to be unique. For each contrast, the `dea` slot stores (when available) metadata describing the DEA, such as the alpha level, LFC threshold, package used to generate the results and its version. The `fea` slot follows the same design. For each contrast, it stores metadata such as the associated DEA contrast from which the FEA was derived, the FEA name, the “shaken” `GeneTonic` output, the tool used to perform the enrichment analysis and its version. Original result objects and feature-level summaries are linked through the `metadata` and `rowData`.

method provides a quick contrast-level overview, such as the number of up/down-regulated genes or enriched terms. To support multi-condition workflows, DeeDeeExperiment provides helper functions for integrating results from limma objects with multiple contrasts (Ritchie *et al.* 2015) and pseudobulk analyses generated with muscat (Crowell *et al.* 2020). As described above, the original DEA results objects are stored once within the metadata and referenced across contrasts. For limma, this means the underlying MArrayLM object is linked rather than duplicated. In contrast, muscat pseudobulk workflows produce separate results tables per contrast, which are stored individually but still organized coherently via the same metadata infrastructure.

Existing common practices based on the original SummarizedExperiment or SingleCellExperiment objects (Martin Morgan 2017, Amezquita *et al.* 2020) combined with ad hoc result lists do not yet provide a unified, contrast-centric container that co-stores DEA and FEA results and integrates feature-level statistics into the rowData across multiple contrasts. Other packages also introduce domain-specific data structures, e.g. MultiAssayExperiment for multi-omics integration (Ramos *et al.* 2017), or the DatasetExperiment class in structToolbox (Lloyd *et al.* 2020) for metabolomics. However, these solutions are designed to organize assay-level data rather than providing a general, contrast-centric S4 framework for storing and managing multiple DEA and FEA analysis results. Our proposed class, DeeDeeExperiment, fills this niche while preserving seamless compatibility with most other Bioconductor tools.

To further enhance reproducibility, contextual information related to specific DEA results can be stored in a DeeDeeExperiment object using the addScenarioInfo() method. This functionality helps document the experimental setup, clarify comparisons, and facilitate downstream interpretation. Looking ahead, such contextual metadata could also be leveraged by large language models (LLMs) to enhance machine-assisted interpretation (Hao *et al.* 2024), opening opportunities for integrating DeeDeeExperiment objects with emerging AI-driven tools. Since DeeDeeExperiment extends the SingleCellExperiment class, it maintains compatibility with the broad ecosystem of Bioconductor tools. We envision extending compatibility to other classes derived from SingleCellExperiment, such as SpatialExperiment (Righelli *et al.* 2022) and TreeSummarizedExperiment (Huang *et al.* 2020) among others, which do not directly inherit from DeeDeeExperiment. To support this, coercion methods could be implemented to allow users to leverage DeeDeeExperiment's full functionality while preserving class-specific slots. In practical settings, this means that once an analysis is completed and a DeeDeeExperiment object is shared, collaborators can readily visualize the data with scater (McCarthy *et al.* 2017) (e.g. tSNE or UMAP) or explore it interactively with iSEE (Rue-Albrecht *et al.* 2018) without performing any extra data handling. A complete documentation and description of use cases for the DeeDeeExperiment package is provided in its vignette (rendered at <https://imbeimainz.github.io/DeeDeeExperiment/>).

3 Conclusion

DeeDeeExperiment provides a robust, standardized and extensible class for managing, and integrating DEA and FEA results. By

storing these results in a conceptually clear manner and capturing the essential details of each analysis within a single container, DeeDeeExperiment enables users to organize, retrieve, explore, contextualize, and interpret their analysis results more efficiently across multiple contrasts, or even across multiple experimental scenarios. Consolidating all information in one data object supports more nuanced and quantitative approaches beyond simple overlap strategies, such as Venn diagrams, enabling trustworthy summaries of complex experimental measurements.

While DeeDeeExperiment is designed to store a wide set of DEA and FEA results, future work will be required to capture additional provenance details such as sessions and environments, ensuring an even higher standard of reproducibility. Being part of the Bioconductor ecosystem, DeeDeeExperiment maintains full compatibility with other existing tools for downstream analyses, promoting reproducibility and elevating individual DE/enrichment results and their metadata as essential component of research output to share not only among collaborators, but also to the broader scientific community. DeeDeeExperiment is available on Bioconductor (<https://bioconductor.org/packages/DeeDeeExperiment>), and the development version is maintained on GitHub (<https://github.com/imbeimainz/DeeDeeExperiment>).

Acknowledgements

The authors thank the Bioconductor community for valuable feedback and suggestions, and also thank Ahmed Hassan, Annkathrin Nedwed, Alicia Schulze, and Alina Jenn for the insightful discussions and constructive suggestions on features and functionality, as well as for testing early versions of the package.

Author contributions

Najla Abassi (Conceptualization [equal], Data curation [equal], Methodology [lead], Software [equal], Validation [equal], Writing—original draft [lead], Writing—review & editing [lead]), Lea Schwarz (Data curation [equal], Methodology [equal], Software [equal], Writing—review & editing [supporting]), Edoardo Filippi (Validation [equal], Writing—review & editing [supporting]), and Federico Marini (Conceptualization [equal], Funding acquisition [lead], Methodology [equal], Project administration [lead], Resources [equal], Software [equal], Supervision [lead], Writing—review & editing [lead])

Supplementary material

Supplementary material is available at *Bioinformatics* online.

Conflict of interests

None declared.

Funding

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Projektnummer 318346496—SFB1292/2 TP19N [to N.A. and F.M.].

References

- Alexa A, Rahnenführer J, Lengauer T. Improved scoring of functional groups from gene expression data by decorrelating graph structure. *Bioinformatics* 2006;**22**:1600–7. <https://doi.org/10.1093/bioinformatics/btl140>
- Amezquita RA, Lun ATL, Becht E *et al.* Publisher correction: orchestrating single-cell analysis with bioconductor. *Nat Methods* 2020;**17**:242. <https://doi.org/10.1038/s41592-019-0700-8>
- Chen C, Wang J, Pan D *et al.* Applications of multi-omics analysis in human diseases. *MedComm (2020)* 2023;**4**:e315. <https://doi.org/10.1002/mco2.315>
- Chen EY, Tan CM, Kou Y *et al.* Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics* 2013;**14**:128. <https://doi.org/10.1186/1471-2105-14-128>
- Crowell HL, Sonesson C, Germain P-L *et al.* muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nat Commun* 2020;**11**:6077. <https://doi.org/10.1038/s41467-020-19894-4>
- Hajjar G, Barros Santos MC, Bertrand-Michel J *et al.* Scaling-up metabolomics: current state and perspectives. *TrAC Trends Anal Chem* 2023;**167**:117225. <https://doi.org/10.1016/j.trac.2023.117225>
- Hao M, Gong J, Zeng X *et al.* Large-scale foundation model on single-cell transcriptomics. *Nat Methods* 2024;**21**:1481–91. <https://doi.org/10.1038/s41592-024-02305-7>
- Huang R, Sonesson C, Ernst FG *et al.* TreeSummarizedExperiment: a S4 class for data with hierarchical structure. *F1000Res* 2020;**9**:1246. <https://doi.org/10.12688/f1000research.26669.2>
- Huber W, Carey VJ, Gentleman R *et al.* Orchestrating high-throughput genomic analysis with bioconductor. *Nat Methods* 2015;**12**:115–21. <https://doi.org/10.1038/nmeth.3252>
- Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci* 2016;**374**:20150202. <https://doi.org/10.1098/rsta.2015.0202>
- Korotkevich G, Sukhov V, Budin N *et al.* Fast gene set enrichment analysis. *bioRxiv*, <https://doi.org/10.1101/060012>, 2021, preprint: not peer reviewed. <https://doi.org/10.1101/060012>
- Lloyd GR, Jankevics A, Weber RJM. struct: an R/bioconductor-based framework for standardized metabolomics data analysis and beyond. *Bioinformatics* 2020;**36**:5551–2. <https://doi.org/10.1093/bioinformatics/btaa1031>
- Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 2014;**15**:550. <https://doi.org/10.1186/s13059-014-0550-8>
- Ludt A, Ustjanzew A, Binder H *et al.* Interactive and reproducible workflows for exploring and modeling RNA-seq data with pcaexplorer, ideal, and genetonic. *Curr Protoc* 2022;**2**:e411. <https://doi.org/10.1002/cpz1.411>
- Maaten L. V. D, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.
- Marini F, Ludt A, Linke J *et al.* Genetonic: an R/bioconductor package for streamlining the interpretation of RNA-seq data. *BMC Bioinformatics* 2021;**22**:610. <https://doi.org/10.1186/s12859-021-04461-5>
- Morgan M, Obenchain V, Hester J *et al.* Summarized Experiment: A container (S4 class) for matrix-like assays. <https://doi.org/10.18129/B9.bioc.SummarizedExperiment>, R package version 1.40.0. 2025. <https://bioconductor.org/packages/SummarizedExperiment>
- McCarthy DJ, Campbell KR, Lun ATL *et al.* Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* 2017;**33**:1179–86. <https://doi.org/10.1093/bioinformatics/btw777>
- McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. 2018. <https://doi.org/10.48550/arXiv.1802.03426>
- Peng H, Wang H, Kong W *et al.* Optimizing differential expression analysis for proteomics data via high-performing rules and ensemble inference. *Nat Commun* 2024;**15**:3922. <https://doi.org/10.1038/s41467-024-47899-w>
- Ramos M, Schiffer L, Re A *et al.* Software for the integration of multiomics experiments in bioconductor. *Cancer Res* 2017;**77**:e39–42. <https://doi.org/10.1158/0008-5472.can-17-0344>
- Righelli D, Weber LM, Crowell HL *et al.* Spatialexperiment: infrastructure for spatially-resolved transcriptomics data in R using bioconductor. *Bioinformatics* 2022;**38**:3128–31. <https://doi.org/10.1093/bioinformatics/btac299>
- Ritchie ME, Phipson B, Wu D *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 2015;**43**:e47. <https://doi.org/10.1093/nar/gkv007>
- Robinson MD, McCarthy DJ, Smyth GK. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 2010;**26**:139–40. <https://doi.org/10.1093/bioinformatics/btp616>
- Rue-Albrecht K, Marini F, Sonesson C *et al.* iSEE: interactive summarized experiment explorer. *F1000Res* 2018;**7**:741. <https://doi.org/10.12688/f1000research.14966.1>
- Smirnov D, Konstantinovskiy N, Prokisch H. Integrative omics approaches to advance rare disease diagnostics. *J Inherit Metab Dis* 2023;**46**:824–38. <https://doi.org/10.1002/jimd.12663>
- Van den Berge K, Hembach KM, Sonesson C *et al.* RNA sequencing data: Hitchhiker's guide to expression analysis. *Annu Rev Biomed Data Sci* 2019;**2**:139–73. <https://doi.org/10.1146/annurev-biodatasci-072018-021255>
- Yu G, Wang L-G, Han Y *et al.* clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* 2012;**16**:284–7. <https://doi.org/10.1089/omi.2011.0118>

© The Author(s) 2026. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Bioinformatics, 2026, 42, 1–4

<https://doi.org/10.1093/bioinformatics/btag157>

Applications Note