



One class to rule them all: **DeeDeeExperiment** for managing and exploring omics analysis results

Najla Abassi

Research Associate, PhD Student

Bioinformatics Division

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

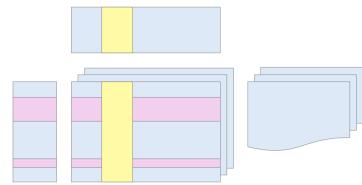
EuroBioC2025, Barcelona, Spain

September 17, 2025



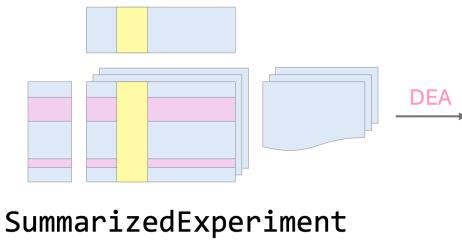
UNIVERSITÄTSmedizin.
MAINZ

Omics data: Too many results, too many tables ...



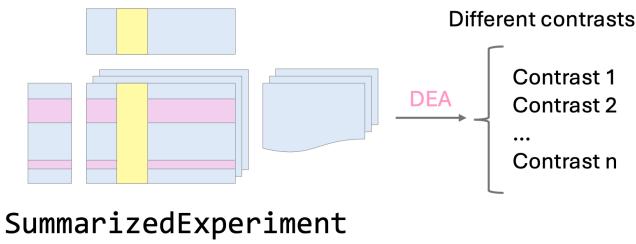
`SummarizedExperiment`

Omics data: Too many results, too many tables ...



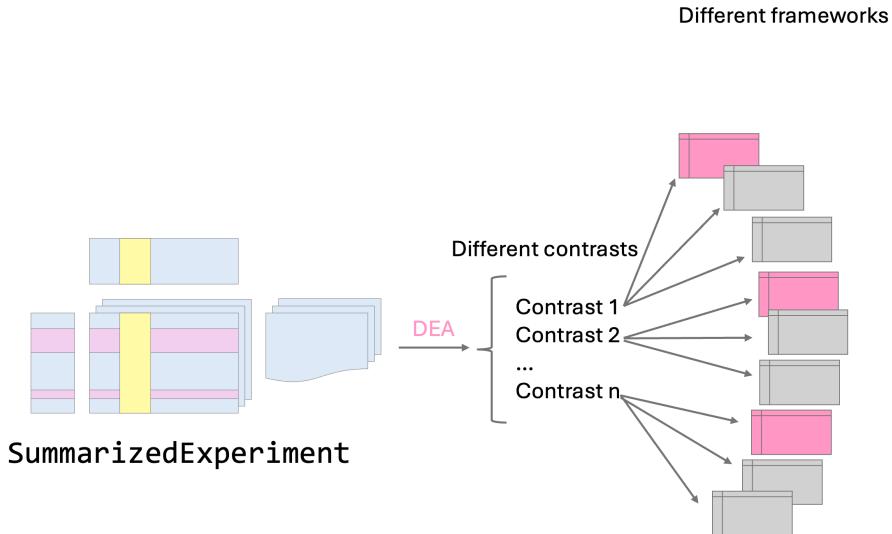
- DEA: Differential Expression Analysis

Omics data: Too many results, too many tables ...



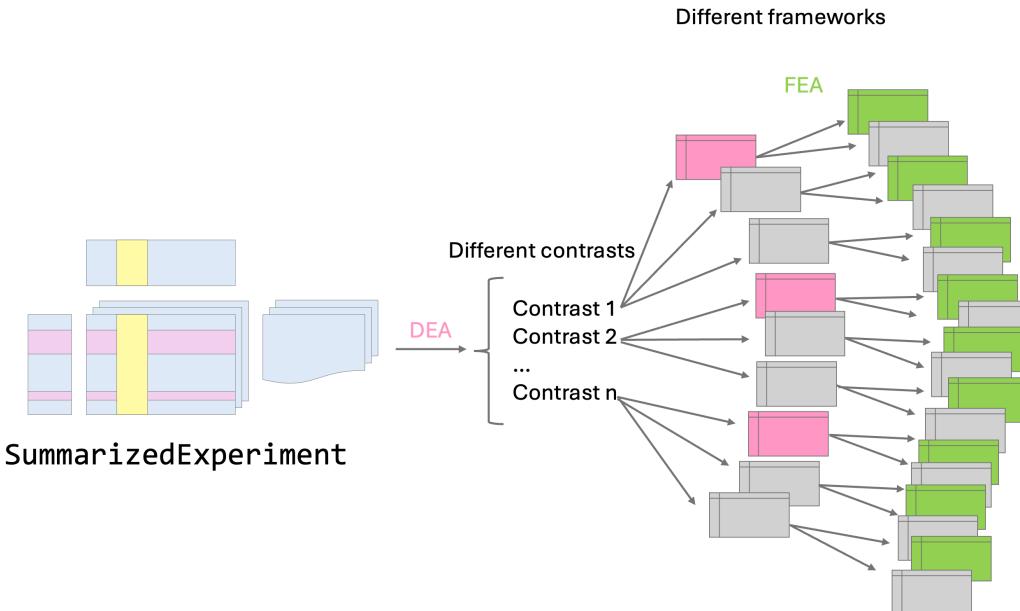
- DEA: Differential Expression Analysis

Omics data: Too many results, too many tables ...



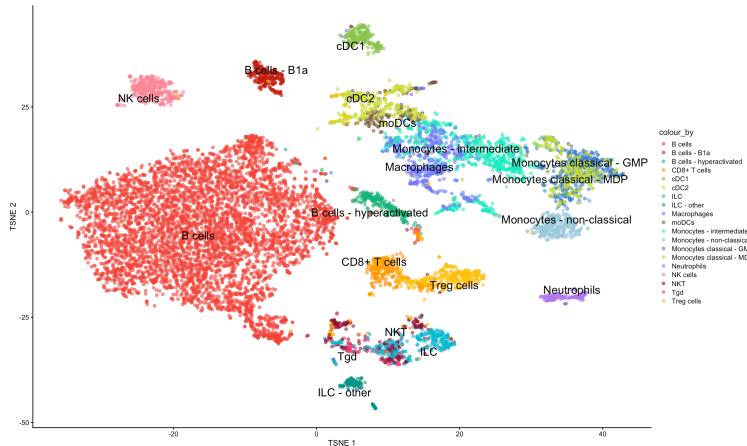
- DEA: Differential Expression Analysis

Omics data: Too many results, too many tables ...



- DEA: Differential Expression Analysis
- FEA: Functional Enrichment Analysis

Problem amplified: Complex designs/Single-cell data ...



Where's the DE Analysis results table from that contrast again??

confused

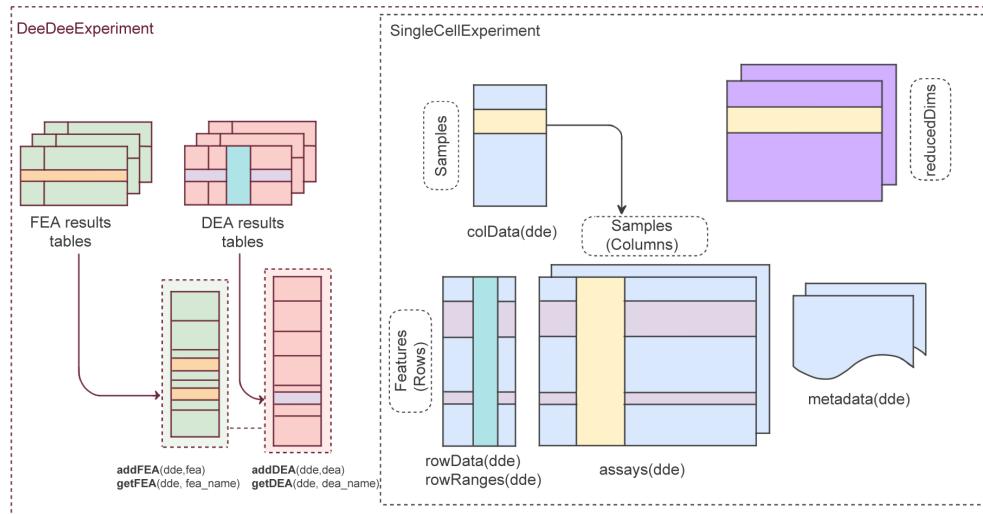


⚠ Problem

- Difficulty to **manage, explore, contextualize** and **reproduce** the results

Our solution

- A new S4 object `DeeDeeExperiment` to structure and store DEA and FEA results in one place, based on the `SingleCellExperiment` object



Note

`DeeDeeExperiment` won't do the analysis for you, it's a **container**

Creating a `dde` object

```
1 # initialize DeeDeeExperiment with dds object and DESeq results (as a named list)
2 dde <- DeeDeeExperiment(sce = dds_macrophage,
3                         de_results = list(
4                           IFNg_vs_naive = IFNg_vs_naive,
5                           Salm_vs_naive = Salm_vs_naive
6                         ))
7
8 dde
class: DeeDeeExperiment
dim: 1000 24
metadata(7): tximetaInfo quantInfo ... txdbInfo version
assays(7): counts abundance ... H cooks
rownames(1000): ENSG00000164741 ENSG00000078808 ... ENSG00000273680
  ENSG00000073536
rowData names(58): gene_id SYMBOL ... Salm_vs_naive_pvalue
  Salm_vs_naive_padj
colnames(24): SAMEA103885102 SAMEA103885347 ... SAMEA103885308
  SAMEA103884949
colData names(15): names sample_id ... condition line
reducedDimNames(0):
mainExpName: NULL
altExpNames(0):
dea(2): IFNg_vs_naive, Salm_vs_naive
fea(0):
```

Note

Supported DE result types: `DESeqResults`, `DGEExact`, `DGELRT`, `MArrayLM`, or `data.frame`

Adding results to a `dde` object

DEA results

```
1 # add results from limma as a MArrayLM object
2 dde <- addDEA(dde, dea = de_limma)
3
4 dde

class: DeeDeeExperiment
dim: 1000 24
metadata(7): tximetaInfo quantInfo ... txdbInfo version
assays(7): counts abundance ... H cooks
rownames(1000): ENSG00000164741 ENSG0000078808 ... ENSG00000273680
  ENSG0000073536
rowData names(61): gene_id SYMBOL ... de_limma_pvalue de_limma_padj
colnames(24): SAMEA103885102 SAMEA103885347 ... SAMEA103885308
  SAMEA103884949
colData names(15): names sample_id ... condition line
reducedDimNames(0):
mainExpName: NULL
altExpNames(0):
dea(3): IFNg_vs_naive, Salm_vs_naive, de_limma
fea(0):

  1 # inspect the columns of the rowData
  2 tail(names(rowData(dde)))

[1] "Salm_vs_naive_log2FoldChange" "Salm_vs_naive_pvalue"
[3] "Salm_vs_naive_padj"          "de_limma_log2FoldChange"
[5] "de_limma_pvalue"            "de_limma_padj"
```

Adding results to a `dde` object

FEA results

→ Can be added to a `dde` object using the `addFEA()` method.

```
1 # add FEA results as a named list
2 dde <- addFEA(dde,
3   fea = list(IFNg_vs_naive = topGO_results_list$ifng_vs_naive))
4
5 # add FEA results as a single object
6 dde <- addFEA(dde, fea = gost_res$result)
7
8 # add FEA results and specify the FEA tool
9 dde <- addFEA(dde, fea = clusterPro_res, fea_tool = "clusterProfiler")
```

FEA results formats that are supported natively within `DeeDeeExperiment`

```
1 supportedfeaformats()
  Format      Package
1 data.frame    topGO
2 enrichResult clusterProfiler
3 gseaResult clusterProfiler
4 fgseaResult    fgsea
5 data.frame    gprofiler2
6 data.frame    enrichR
7 data.frame    DAVID
8 data.frame    GeneTonic
```

Linking DE and FE Analysis in a `dde` object

```
1 dde <- linkDEAandFEA(dde,
2                           dea_name = "IFNg_vs_naive",
3                           fea_name = c("IFNg_vs_naive", "gost_res$result"))
```

```
✓ Assigning DEA: "IFNg_vs_naive" to FEA "IFNg_vs_naive"
✓ Assigning DEA: "IFNg_vs_naive" to FEA "gost_res$result"
```

Adding contextual information

```
1 dde <- addScenarioInfo(dde,  
2                           dea_name = "IFNg_vs_naive",  
3                           info = "This results contains the output of a Differential Expression Analysis performed on data from the `macrophage` p  
4 )
```

- **Document experimental setup** for clarity & reproducibility
- **Clarify comparisons** between conditions
- **Provide extra context** that can assist interpretation (e.g. by LLM)

Summary of a `dde` object

Set FDR threshold

```
1 # specify FDR threshold for subsetting DE genes based on adjusted p-values
2 summary(dde,
3           FDR = 0.01)
```

DE Results Summary:

	DEA_name	Up	Down	FDR
IFNg_vs_naive	30	15	0.01	
Salm_vs_naive	80	29	0.01	
de_limma	203	244	0.01	
same_contrast	203	244	0.01	

FE Results Summary:

FEA_Name	Linked_DE	FE_Type	Term_Number
IFNg_vs_naive	IFNg_vs_naive	topGO	955
gost_res\$result	IFNg_vs_naive	gProfiler	2095
ifng_vs_naive	.	clusterProfiler	20
salmonella_vs_naive	.	clusterProfiler	11

Summary of a `dde` object

Display scenario information

```
1 # show contextual information, if available
2 summary(dde,
3   show_scenario_info = TRUE)
```

DE Results Summary:

	DEA_name	Up	Down	FDR
IFNg_vs_naive	36	17	0.05	
Salm_vs_naive	90	34	0.05	
	de_limma	265	303	0.05
	same_contrast	265	303	0.05

FE Results Summary:

FEA_Name	Linked_DE	FE_Type	Term_Number
IFNg_vs_naive	IFNg_vs_naive	topGO	955
gost_res\$result	IFNg_vs_naive	gProfiler	2095
ifng_vs_naive	.	clusterProfiler	20
salmonella_vs_naive	.	clusterProfiler	11

Scenario Info:

- IFNg_vs_naive :
This results contains the output of a Differential Expression Analysis
performed on data from the `macrophage` package, more precisely contrasting
the counts from naive macrophage to those associated with IFNg.

No scenario info for: Salm_vs_naive, de_limma, same_contrast

Accessing results stored in a `dde` object

Accessing DEA results, by DEA name

```
1 # access specific DEA by name, in minimal format
2 knitr::kable(head(getDEA(dde,
3                         dea_name = "Salm_vs_naive")))
```

	Salm_vs_naive_log2FoldChange	Salm_vs_naive_pvalue	Salm_vs_naive_padj
ENSG00000164741	0.0114691	0.9996325	1.0000000
ENSG00000078808	0.7893601	0.9975592	1.0000000
ENSG00000251034	1.1881497	0.2269175	0.9820000
ENSG00000162676	0.4420821	0.8432117	1.0000000
ENSG00000170356	-0.6249201	0.7269723	1.0000000
ENSG00000204257	-0.7293810	0.8838883	1.0000000

Accessing results stored in a `dde` object

Accessing DEA results, as a `list`

```
1 # get dea results as a list, (default: minimal format)
2 lapply(getDEAList(dde), head)
```

```
$IFNg_vs_naive
  log2FoldChange      pvalue      padj
ENSG00000164741  0.22914866 9.805414e-01 1.000000e+00
ENSG00000078808 -0.01153364 1.000000e+00 1.000000e+00
ENSG00000251034 -0.11670132 9.338990e-01 1.000000e+00
ENSG00000162676  0.26914813 9.008170e-01 1.000000e+00
ENSG00000170356 -0.12786371 9.728148e-01 1.000000e+00
ENSG00000204257  4.05502439 1.672976e-56 8.36488e-54
```

```
$Salm_vs_naive
  log2FoldChange      pvalue      padj
ENSG00000164741  0.0114691 0.9996325 1.0000000
ENSG00000078808  0.7893601 0.9975592 1.0000000
ENSG00000251034  1.1881497 0.2269175 0.9820422
ENSG00000162676  0.4420821 0.8432117 1.0000000
ENSG00000170356 -0.6249201 0.7269723 1.0000000
ENSG00000204257 -0.7293810 0.8838883 1.0000000
```

```
$de_limma
  log2FoldChange      pvalue      padj
ENSG00000164741 -0.07694397 8.044824e-01 8.540153e-01
ENSG00000078808  0.78155238 1.672966e-08 3.510702e-07
ENSG00000251034  1.32914271 1.704072e-02 3.215229e-02
ENSG00000204257  0.65000000 1.042010e-01 1.510000e-01
```

Accessing FEA results

`getFEA()` directly accesses specific FEA results.

`getFEAList()` returns all FEAs stored in a `dde` object. Optionally, if `dea_name` is set, it returns all FEAs linked to a specific DEA.

Other operations ...

Renaming results in a `dde` object

Renaming DEA results

```
1 # rename dea, one element
2 dde <- renameDEA(dde,
3                     old_name = "de_limma",
4                     new_name = "ifng_vs_naive_&salm_vs_naive")
```

✓ Renamed DEA entries: "de_limma" to "ifng_vs_naive_&salm_vs_naive"

```
1 # multiple entries at once
2 dde <- renameDEA(dde,
3                     old_name = c("same_contrast", "Salm_vs_naive"),
4                     new_name = c("same_contrast_new", "Salm_vs_naive_new")
5                     )
```

✓ Renamed DEA entries: "same_contrast" and "Salm_vs_naive" to "same_contrast_new" and "Salm_vs_naive_new"



To rename FEAs stored in a `dde` object, use `renameFEA()`

Removing results in a `dde` object

Removing DEA results

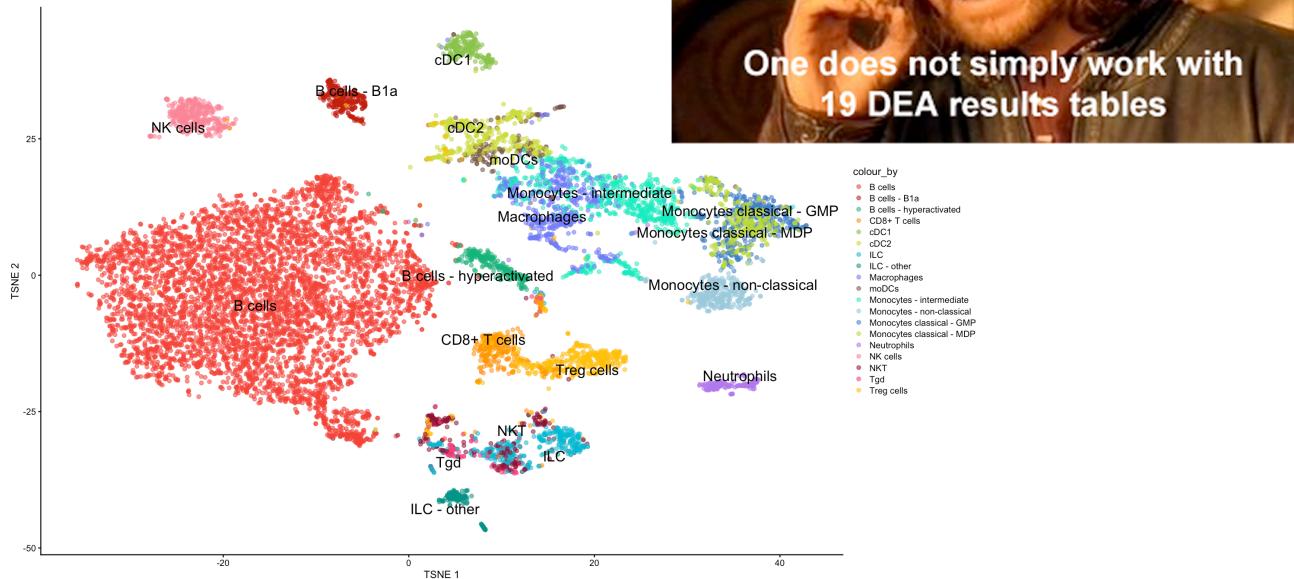
```
1 # removing dea
2 dde <- removeDEA(dde,
3                      c("ifng_vs_naive_&_salm_vs_naive",
4                        "same_contrast_new",
5                        "Salm_vs_naive_new"))
```



Tip

To remove FEAs stored in a `dde` object, use `removeFEA()`

Single-cell in practice

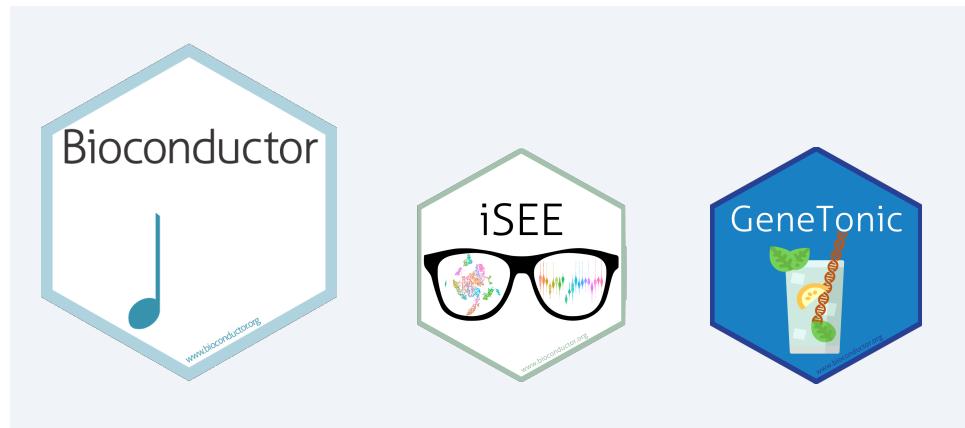


Single-cell in practice (`muscat::pbDS()` → `DeeDeeExperiment`)

```
1 # create res, a list to hold pseudobulk DE results for all contrasts
2 for (i in names(contrast)) {
3   cat("Contrast: ", i, "\n")
4   res <- pbDS(pb,
5             design = mm,
6             contrast = contrast[[i]],
7             verbose = TRUE,
8             BPPARAM = BiocParallel::MulticoreParam(6))
9
10  results_list[[i]] <- res
11 }
12
13 # extract contrast
14 contrast_vtp_DMSO <- res$table$`VTP-DMSO`
15
16 # renaming columns
17 for (cell in names(contrast_vtp_DMSO)) {
18   contrast_vtp_DMSO[[cell]] <-
19   contrast_vtp_DMSO[[cell]] |>
20   dplyr::rename(log2FoldChange = logFC,
21                 pvalue = p_val,
22                 padj = p_adj.loc)
23
24   rownames(contrast_vtp_DMSO[[cell]]) <- contrast_vtp_DMSO[[cell]]$gene
25 }
26
27 # optional: update de + enrich list names
28 new_names <- c(
29   "NK1_A+B" = "NK1_A_B",
30   "NK1_C" = "NK1_C"
```

Downstream operations with `DeeDeeExperiment` objects

`DeeDeeExperiment` objects can be used with other Bioconductor tools like [iSEE](#) & [GeneTonic](#), or anything that works for a SCE object



DeeDeeExperiment is available on Bioconductor



About Learn Packages Developers

Search

Get Started >

Home > Bioconductor 3.22 > Software Packages > **DeeDeeExperiment (development version)**

DeeDeeExperiment

This is the **development** version of DeeDeeExperiment; to use it, please install the [devel version](#) of Bioconductor.

DeeDeeExperiment: An S4 Class for managing and exploring omics analysis results

platforms all rank 2313 / 2320 support 0 / 0 in Bioc devel only build ok updated < 1 week dependencies 68

DOI: [10.18129/B9.bioc.DeeDeeExperiment](https://doi.org/10.18129/B9.bioc.DeeDeeExperiment)

Bioconductor version: Development (3.22)

DeeDeeExperiment is an S4 class extending the SingleCellExperiment class, designed to integrate and manage omics analysis results. It introduces two dedicated slots to store Differential Expression Analysis (DEA) results and Functional Enrichment Analysis (FEA) results, providing a structured approach for downstream analysis.

Author: Najla Abassi [aut, cre] , Lea Rothörl [aut] , Federico Marini [aut]

Maintainer: Najla Abassi <abassi.nejla96@gmail.com>

<https://bioconductor.org/packages/DeeDeeExperiment>

Thank you :)



DeeDeeExperiment on Bioconductor

- *Bioinformatics group @IMBEI*
- *Bioconductor community, developers-forum*