

# Package ‘GraphExperiment’

May 2, 2026

**Title** S4 Class for Quantitative Data and Associated Networks

**Version** 1.1.0

**Date** 2026-03-01

**Description** GraphExperiment provides users and developers with an S4 class that extends `SingleCellExperiment` by offering infrastructure to store and retrieve networks (`igraph` objects) representing how features are associated with each other. The class was designed to store networks inferred from high-dimensional quantitative data, including gene coexpression networks (GCNs), gene regulatory networks (GRNs), and co-abundance networks (from proteomics and metabolomics), as well as networks inferred from other types of data (e.g., protein-protein interactions).

**License** GPL-3

**URL** <https://github.com/almeidasilvaf/GraphExperiment>

**BugReports** <https://support.bioconductor.org/tag/GraphExperiment>

**Encoding** UTF-8

**LazyData** false

**biocViews** DataRepresentation, DataImport, Infrastructure,  
GeneExpression, Transcriptomics, Network, SingleCell

**Depends** SingleCellExperiment, igraph

**Imports** methods, SummarizedExperiment, BiocBaseUtils, S4Vectors

**Suggests** knitr, BiocStyle, testthat, rmarkdown, covr, sessioninfo

**Config/testthat/edition** 3

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/GraphExperiment>

**git\_branch** devel

**git\_last\_commit** 45c7264

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-05-01

**Author** Fabricio Almeida-Silva [aut, cre] (ORCID: <https://orcid.org/0000-0002-5314-2964>),  
Yves Van de Peer [aut] (ORCID: <https://orcid.org/0000-0003-4327-3730>)

**Maintainer** Fabricio Almeida-Silva <fabricio\_almeidasilva@hotmail.com>

## Contents

GraphExperiment-class . . . . .	2
GraphExperiment-coerce . . . . .	3
GraphExperiment-methods . . . . .	4
GraphExperiment-subset . . . . .	6
<b>Index</b>	<b>8</b>

---

GraphExperiment-class GraphExperiment *S4 class*

---

## Description

The GraphExperiment class was designed to represent rectangular, quantitative data (e.g., from transcriptomics, proteomics, metabolomics) along with graphs showing how features (e.g., genes, proteins, compounds) interact with each other. It extends SingleCellExperiment by providing users with an additional slot where graphs can be stored.

## Usage

```
GraphExperiment(..., graphs = list())
```

## Arguments

...	Arguments passed to the SingleCellExperiment constructor function.
graphs	A list of igraph objects with one or multiple graphs. Node names (i.e., $V(\text{graph})\$name$ ) must match rownames of assays.

## Details

Like SingleCellExperiment, the GraphExperiment S4 class stores quantitative data with associated metadata (i.e., rowData and colData) along with embeddings from dimensionality reduction techniques. However, it provides users with an additional container for igraph objects containing graphs describing how features interact with each other. Graphs are stored in a graphs slot, which can hold one or multiple igraph objects with some sort of network representation of the features in rownames. Example graphs can be coexpression networks, regulatory networks, or co-abundance networks.

Importantly, node names in each graph must match rownames of the assays, and subsetting methods simultaneously subset assays, rowData, and graphs.

Besides the constructor function (GraphExperiment()), a GraphExperiment object can also be created by coercing from a SummarizedExperiment or SingleCellExperiment object.

## Value

A GraphExperiment object.

**Examples**

```

# Example 1: from constructor function ----
## Simulate a matrix with 200 genes and 100 cells
gene_ids <- paste0("gene", seq_len(200))
cell_ids <- paste0("cell", seq_len(100))
mat <- matrix(rpois(20000, 5), ncol = 100, dimnames = list(gene_ids, cell_ids))

## Create a graph from correlations (`igraph` object)
g <- graph_from_adjacency_matrix(cor(t(mat)), weighted = TRUE)

## Construct `GraphExperiment` object
ge <- GraphExperiment(
  assays = list(counts = mat),
  graphs = list(cor = g)
)
ge

# Example 2: From `SingleCellExperiment` object ----
sce <- SingleCellExperiment(assays = list(counts = mat))
ge <- as(sce, "GraphExperiment")
ge

```

---

GraphExperiment-coerce

*Coercion methods for GraphExperiment objects*


---

**Description**

The GraphExperiment class inherits from SingleCellExperiment, which in turn inherits from (Ranged)SummarizedExperiment. To ensure GraphExperiment easily interoperates with these classes, we provide users with traditional as() coercion methods.

**Arguments**

from                    An object of class SingleCellExperiment, SummarizedExperiment, or RangedSummarizedExperiment to be coerced to GraphExperiment.

**Value**

A GraphExperiment object.

**Examples**

```

# Simulate a count matrix
gene_ids <- paste0("gene", seq_len(200))
cell_ids <- paste0("cell", seq_len(100))
mat <- matrix(rpois(20000, 5), ncol = 100, dimnames = list(gene_ids, cell_ids))

# Coerce from `SummarizedExperiment`
se <- SummarizedExperiment(assays = list(mat))
as(se, "GraphExperiment")

```

```
# Coerce from `RangedSummarizedExperiment`
rse <- as(se, "RangedSummarizedExperiment")
as(rse, "GraphExperiment")

# Coerce from `SingleCellExperiment`
sce <- as(se, "SingleCellExperiment")
as(sce, "GraphExperiment")
```

---

## GraphExperiment-methods

### *Methods for GraphExperiment objects*

---

## Description

The `GraphExperiment` class provides users with methods to get and set graphs (igraph objects) representing how features of `SingleCellExperiment` objects relate to each other.

## Usage

```
## S4 method for signature 'GraphExperiment'
rowData(x, use.names = TRUE, ...)

## S4 method for signature 'GraphExperiment'
graphs(x)

## S4 method for signature 'GraphExperiment,missing'
graph(x, i)

## S4 method for signature 'GraphExperiment,ANY'
graph(x, i)

## S4 method for signature 'GraphExperiment'
graphNames(x)

## S4 replacement method for signature 'GraphExperiment'
graphs(x) <- value

## S4 replacement method for signature 'GraphExperiment'
graph(x, i) <- value

## S4 replacement method for signature 'GraphExperiment,character'
graphNames(x) <- value
```

## Arguments

<code>x</code>	A <code>GraphExperiment</code> object.
<code>use.names</code>	Passed to the <code>rowData</code> method of <code>SingleCellExperiment</code> . Default: <code>TRUE</code> .
<code>...</code>	Ignored.
<code>i</code>	List element (numeric for index, character for name) of the element to access or replace.
<code>value</code>	Replacement value for replacement methods.

**Value**

Return values depend on the method. See details and examples.

**graphs and graph methods**

`graphs(x)`: Getter for a SimpleList of igraph objects.

`graphs(x) <- value`: Setter for a SimpleList or list (coerced to SimpleList) of igraph objects.

`graph(x, i)`: Getter for an igraph object containing graph `i` from the list stored in `graphs`.

`graph(x, i) <- value`: Setter for an igraph object to be stored in element `i` of the list `graphs`

**graphNames methods**

`graphNames(x)`: Getter to extract names of graphs in the `graphs` slot.

`graphNames(x) <- value`: Setter to assign new names to the graphs stored in the `graphs` slot.

**rowData method**

`rowData(x)`: Getter to extract `rowData` (as in `SingleCellExperiment` objects), but with node attributes of graphs included.

**Examples**

```
# Simulate elements of a GraphExperiment object
## Assays
gene_ids <- paste0("gene", seq_len(200))
cell_ids <- paste0("cell", seq_len(100))
mat <- matrix(rpois(20000, 5), ncol = 100, dimnames = list(gene_ids, cell_ids))

## Graph (with node attributes)
g <- graph_from_adjacency_matrix(cor(t(mat)), weighted = TRUE)
V(g)$degree <- igraph::strength(g)

## rowData
rdata <- data.frame(
  row.names = gene_ids,
  pathway = sample(c("P1", "P2"), size = length(gene_ids), replace = TRUE),
  coding = sample(c(TRUE, FALSE), size = length(gene_ids), replace = TRUE)
)

# Create a GraphExperiment object
ge <- GraphExperiment(
  assays = list(counts = mat),
  rowData = rdata,
  graphs = list(cor = g)
)
ge

# Extract graph names
graphNames(ge)
ge

# Extract graphs
graphs(ge)
```

```

graph(ge, "cor")

# Add a new graph
graph(ge, "newcor") <- g
ge

# Add a list of graphs
graphs(ge) <- list(cor = g, newcor = g)
ge

# Replace graph names
graphNames(ge) <- c("network1", "network2")
ge

# Access rowData (note: rowData + node attributes combined)
rowData(ge)

```

---

GraphExperiment-subset

*Subsetting GraphExperiment objects*


---

## Description

The subsetting method for [GraphExperiment](#) objects ensures that nodes from igraph objects are filtered to match rows and columns with the remainder of the object.

## Arguments

x	A <a href="#">GraphExperiment</a> object.
i	Numeric, row indices for subsetting.
j	Numeric, column indices for subsetting.

## Value

a [GraphExperiment](#) object.

## subset

[**:** subsetting method

## Examples

```

# Simulate elements of a GraphExperiment object
## Assays
gene_ids <- paste0("gene", seq_len(200))
cell_ids <- paste0("cell", seq_len(100))
mat <- matrix(rpois(20000, 5), ncol = 100, dimnames = list(gene_ids, cell_ids))

## Graph (with node attributes)
g <- graph_from_adjacency_matrix(cor(t(mat)), weighted = TRUE)
V(g)$degree <- igraph::strength(g)

```

```
## rowData
rdata <- data.frame(
  row.names = gene_ids,
  pathway = sample(c("P1", "P2"), size = length(gene_ids), replace = TRUE),
  coding = sample(c(TRUE, FALSE), size = length(gene_ids), replace = TRUE)
)

# Create a GraphExperiment object
ge <- GraphExperiment(
  assays = list(counts = mat),
  rowData = rdata,
  graphs = list(cor = g)
)
ge

# Subset object
ge[1:5, 1:5]
```

# Index

[, GraphExperiment, ANY, ANY, ANY-method  
(GraphExperiment-subset), 6

coerce, RangedSummarizedExperiment, GraphExperiment-method  
(GraphExperiment-coerce), 3

coerce, SingleCellExperiment, GraphExperiment-method  
(GraphExperiment-coerce), 3

coerce, SummarizedExperiment, GraphExperiment-method  
(GraphExperiment-coerce), 3

graph (GraphExperiment-methods), 4

graph, GraphExperiment, ANY-method  
(GraphExperiment-methods), 4

graph, GraphExperiment, missing-method  
(GraphExperiment-methods), 4

graph<- (GraphExperiment-methods), 4

graph<-, GraphExperiment-method  
(GraphExperiment-methods), 4

GraphExperiment, 6

GraphExperiment  
(GraphExperiment-class), 2

GraphExperiment-class, 2

GraphExperiment-coerce, 3

GraphExperiment-methods, 4

GraphExperiment-subset, 6

graphNames (GraphExperiment-methods), 4

graphNames, GraphExperiment-method  
(GraphExperiment-methods), 4

graphNames<- (GraphExperiment-methods),  
4

graphNames<-, GraphExperiment, character-method  
(GraphExperiment-methods), 4

graphs (GraphExperiment-methods), 4

graphs, GraphExperiment-method  
(GraphExperiment-methods), 4

graphs<- (GraphExperiment-methods), 4

graphs<-, GraphExperiment-method  
(GraphExperiment-methods), 4

rowData (GraphExperiment-methods), 4

rowData, GraphExperiment-method  
(GraphExperiment-methods), 4