

# Package ‘gedi2’

October 5, 2026

**Type** Package

**Title** Gene Expression Decomposition and Integration

**Version** 2.3.6

**Date** 2026-10-02

**Description** A memory-efficient implementation for integrating gene expression data from single-cell RNA sequencing experiments. Uses a C++ backend with thin R wrappers to enable analysis of large-scale single-cell datasets. The package supports multiple data modalities including count matrices, paired data (splicing, RNA velocity, CITE-seq), and binary indicators. It implements a latent variable model with block coordinate descent optimization for dimensionality reduction and batch effect correction. The method is described in Mikaeili Namini et al. (2026) <[doi:10.1093/bioinformatics/btag334](https://doi.org/10.1093/bioinformatics/btag334)>, building on the original GEDI model of Madrigal et al. (2024) <[doi:10.1038/s41467-024-50963-0](https://doi.org/10.1038/s41467-024-50963-0)>.

**License** MIT + file LICENSE

**URL** <https://github.com/csglab/gedi2>

**BugReports** <https://github.com/csglab/gedi2/issues>

**Depends** R (>= 4.0.0)

**Imports** Rcpp (>= 1.0.0), R6 (>= 2.5.0), Matrix (>= 1.3.0), ggplot2, scales, methods, stats, utils

**LinkingTo** Rcpp, RcppEigen

**Suggests** hdf5r, uwot, digest, glmnet, Seurat, SeuratObject, SingleCellExperiment, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**SystemRequirements** GNU make

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**NeedsCompilation** yes

**Author** Arsham Mikaeili Namini [aut, cre],  
Hamed S.Najafabadi [aut]

**Maintainer** Arsham Mikaeili Namini <arsham.mikaeilinamini@mail.mcgill.ca>  
**Repository** CRAN  
**Date/Publication** 2026-10-05 09:50:02 UTC

**Contents**

|                                         |           |
|-----------------------------------------|-----------|
| check_optional_dependencies . . . . .   | 2         |
| CreateGEDIOObject . . . . .             | 3         |
| gedi_to_seurat . . . . .                | 5         |
| install_optional_dependencies . . . . . | 6         |
| list_h5_structure . . . . .             | 7         |
| plot_convergence . . . . .              | 8         |
| plot_dispersion . . . . .               | 9         |
| plot_embedding . . . . .                | 10        |
| plot_features . . . . .                 | 12        |
| plot_feature_ratio . . . . .            | 14        |
| plot_vector_field . . . . .             | 15        |
| read_h5 . . . . .                       | 17        |
| read_h5ad . . . . .                     | 18        |
| seurat_to_gedi . . . . .                | 19        |
| write_h5ad . . . . .                    | 21        |
| <b>Index</b>                            | <b>23</b> |

---

|                                         |
|-----------------------------------------|
| check_optional_dependencies             |
| <i>Check GEDI Optional Dependencies</i> |

---

**Description**

Check which optional dependencies are installed and display their status.

**Usage**

check\_optional\_dependencies()

**Value**

Named logical vector indicating which optional packages are installed

**Examples**

check\_optional\_dependencies()

---

CreateGEDIObject*Create GEDI Object*

---

## Description

Creates and configures a GEDI (Gene Expression Data Integration) model object. This implementation uses Option 2 memory optimization: C++ computes  $Y_i$  from  $M$ , eliminating duplicate  $Y_i$  storage in R.

## Usage

```
CreateGEDIObject(  
  Samples,  
  M = NULL,  
  Y = NULL,  
  X = NULL,  
  colData = NULL,  
  C = NULL,  
  H = NULL,  
  K = 40,  
  mode = "Bsphere",  
  adjustD = TRUE,  
  orthoZ = TRUE,  
  Z_shrinkage = 1,  
  A_shrinkage = 1,  
  Qi_shrinkage = 1,  
  Rk_shrinkage = 1,  
  oi_shrinkage = 1,  
  o_shrinkage = 1,  
  si_shrinkage = 1,  
  fixed_si = NA,  
  rsvd_p = 10,  
  rsvd_sdist = "normal",  
  verbose = 1,  
  num_threads = 1  
)
```

## Arguments

|         |                                                                                                                                                         |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Samples | Factor or character vector indicating sample of origin for each cell                                                                                    |
| M       | Raw count matrix (sparse or dense), or list of two matrices for paired data. C++ will compute $Y_i = \log(M+1)$ internally - no $Y_i$ copy stored in R! |
| Y       | Log-transformed expression matrix (optional if M provided)                                                                                              |
| X       | Binary indicator matrix (optional if M or Y provided)                                                                                                   |
| colData | Optional data.frame with cell metadata                                                                                                                  |

|                                                                                               |                                                                                     |
|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| C                                                                                             | Gene-level prior matrix (genes x pathways)                                          |
| H                                                                                             | Sample-level covariate matrix (covariates x samples)                                |
| K                                                                                             | Number of latent factors (default: 10)                                              |
| mode                                                                                          | Normalization mode: "BI2" or "Bsphere" (default: "BI2")                             |
| adjustD                                                                                       | Whether to adjust D based on B row norms (default: TRUE)                            |
| orthoZ                                                                                        | Whether Z columns should be orthogonal (default: TRUE)                              |
| Z_shrinkage, A_shrinkage, Qi_shrinkage, Rk_shrinkage, oi_shrinkage, o_shrinkage, si_shrinkage | Regularization strengths (default: 1)                                               |
| fixed_si                                                                                      | Fix cell library sizes at this value, or NA to optimize (default: NA)               |
| rsvd_p                                                                                        | Oversampling parameter for randomized SVD (default: 10)                             |
| rsvd_sdist                                                                                    | Random distribution for rSVD: "normal", "unif", or "rademacher" (default: "normal") |
| verbose                                                                                       | Verbosity level: 0 (silent), 1 (info), 2 (debug) (default: 1)                       |
| num_threads                                                                                   | Number of OpenMP threads (default: 0 = auto)                                        |

## Value

GEDI R6 object with memory-efficient architecture

## Examples

```
# Load example data
pbmc_small <- SeuratObject::pbmc_small

# Basic usage - memory efficient!
model <- CreateGEDIOObject(
  Samples = pbmc_small@meta.data$orig.ident,
  M = pbmc_small@assays$RNA@counts, # Only M stored in R; Yi computed in C++
  K = 10,
  num_threads = 1
)

# Train the model
model$train(iterations = 50)

# Access results via active bindings
Z <- model$Z
params <- model$params

# Access projections (computed on demand, cached)
zdb <- model$projections$ZDB
db <- model$projections$DB

# Access imputed expression
Y_imputed <- model$imputed$Y() # No M needed!
Y_var <- model$imputed$variance(pbmc_small@assays$RNA@counts) # M required
```

gedi\_to\_seurat

*Convert GEDI Model to Seurat Object***Description**

Creates a Seurat object from a trained GEDI model, including imputed data, projections, and embeddings.

**Usage**

```
gedi_to_seurat(
  model,
  M = NULL,
  project = "GEDI",
  assay = "RNA",
  use_imputed = TRUE,
  add_projections = TRUE,
  add_embeddings = TRUE,
  min_cells = 0,
  min_features = 0,
  verbose = TRUE
)
```

**Arguments**

|                 |                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------|
| model           | GEDI model object (trained)                                                                                        |
| M               | Original count matrix (optional). If not provided, will use back-transformed imputed values as approximate counts. |
| project         | Character, project name for Seurat object (default: "GEDI")                                                        |
| assay           | Character, name for the main assay (default: "RNA")                                                                |
| use_imputed     | Logical, whether to add imputed data as separate assay (default: TRUE)                                             |
| add_projections | Logical, whether to add ZDB and DB projections as separate assays (default: TRUE)                                  |
| add_embeddings  | Logical, whether to add UMAP and PCA embeddings if available (default: TRUE)                                       |
| min_cells       | Integer, filter genes with counts in < min_cells (default: 0)                                                      |
| min_features    | Integer, filter cells with < min_features genes (default: 0)                                                       |
| verbose         | Logical, whether to print progress messages (default: TRUE)                                                        |

**Value**

Seurat object with:

- RNA assay: Original or back-transformed counts

- imputed assay: GEDI imputed expression (if use\_imputed = TRUE)
- ZDB/DB/ADB assays: GEDI projections (if add\_projections = TRUE)
  - ZDB: Batch-corrected gene expression (genes x cells)
  - DB: Cell embeddings in latent factor space (K x cells)
  - ADB: Pathway activities (pathways x cells) - only if C matrix provided
- umap/pca reductions: Embeddings (if add\_embeddings = TRUE and cached)
- meta.data: Sample labels and colData from GEDI model

## Examples

```
# Load example data
pbmc <- SeuratObject::pbmc_small

# Train GEDI model
gedi_model <- seurat_to_gedi(pbmc, K = 15)
gedi_model$train(iterations = 10)

# Convert back to Seurat
seurat_obj <- gedi_to_seurat(
  gedi_model,
  use_imputed = TRUE,
  add_projections = TRUE,
  add_embeddings = TRUE
)

# Now can use Seurat functions
library(Seurat)
DefaultAssay(seurat_obj) <- "imputed"
seurat_obj <- FindVariableFeatures(seurat_obj)
```

---

install\_optional\_dependencies

*List Optional GEDI Dependencies*

---

## Description

Reports which optional packages are needed and provides install commands. Does **not** install anything automatically.

## Usage

```
install_optional_dependencies(which = "all", verbose = TRUE)
```

**Arguments**

**which** Character vector specifying which dependency groups to query. Options: "h5" (hdf5r), "umap" (uwot), "validation" (digest), or "all" (default).

**verbose** Logical, whether to print messages (default: TRUE)

**Value**

A named logical vector indicating which packages are installed (invisibly).

**Examples**

```
# Show which optional packages are missing
install_optional_dependencies()
```

---

|                   |                                          |
|-------------------|------------------------------------------|
| list_h5_structure | <i>List structure of H5 or H5AD file</i> |
|-------------------|------------------------------------------|

---

**Description**

Helper function to explore H5/H5AD file structure.

**Usage**

```
list_h5_structure(file_path, recursive = TRUE)
```

**Arguments**

**file\_path** Character. Path to the H5 or H5AD file.

**recursive** Logical. List all nested groups. Default TRUE.

**Value**

data.frame with file structure information

**Examples**

```
# Round-trip: write a tiny H5AD via write_h5ad() then list its structure.
if (requireNamespace("hdf5r", quietly = TRUE) &&
    requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)
```

```
tmp <- tempfile(fileext = ".h5ad")
write_h5ad(model, tmp)
list_h5_structure(tmp)
unlink(tmp)
}
```

---

|                  |                                  |
|------------------|----------------------------------|
| plot_convergence | <i>Plot Training Convergence</i> |
|------------------|----------------------------------|

---

**Description**

Visualizes convergence of model parameters during training. Supports multiple layout styles for different use cases.

**Usage**

```
plot_convergence(
  model,
  layout = c("faceted", "separate", "compact"),
  params = NULL,
  log_scale = TRUE,
  smooth = FALSE,
  title = "Training Convergence"
)
```

**Arguments**

|           |                                                                                                                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| model     | GEDI model object                                                                                                                                                                                                                                                                    |
| layout    | Character, layout style: <ul style="list-style-type: none"><li>"faceted": Single plot with facet_wrap (default, best for reports)</li><li>"separate": List of individual plots (for interactive exploration)</li><li>"compact": Two-panel plot (global vs sample-specific)</li></ul> |
| params    | Character vector, which parameters to include. Options: "Z", "A", "o", "Bi", "Qi", "oi", "si", "Rk", "Ro", "sigma2". NULL means all available.                                                                                                                                       |
| log_scale | Logical, use log10 scale for y-axis                                                                                                                                                                                                                                                  |
| smooth    | Logical, add smooth trend line                                                                                                                                                                                                                                                       |
| title     | Character, plot title                                                                                                                                                                                                                                                                |

**Value**

ggplot2 object (for "faceted" or "compact") or list of ggplot2 objects (for "separate")

**Examples**

```

if (requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 10, track_interval = 2)
  plot_convergence(model)
  plots <- plot_convergence(model, layout = "separate")
}

```

---

|                 |                                 |
|-----------------|---------------------------------|
| plot_dispersion | <i>Plot Dispersion Analysis</i> |
|-----------------|---------------------------------|

---

**Description**

Visualizes the relationship between expected and observed variance for count data. Useful for assessing model fit quality.

**Usage**

```

plot_dispersion(
  dispersion_df,
  show_identity = TRUE,
  point_size = 0.1,
  alpha = 0.5,
  title = "Dispersion Analysis"
)

```

**Arguments**

|               |                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------|
| dispersion_df | Data frame from compute_dispersion() with columns: Expected_Var, Observed_Var, Sample, n, bin |
| show_identity | Logical, whether to show y=x identity line                                                    |
| point_size    | Numeric, size of points                                                                       |
| alpha         | Numeric, transparency of points                                                               |
| title         | Character, plot title                                                                         |

**Value**

ggplot2 object

**Examples**

```

if (requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  M <- pbmc_small@assays$RNA@counts
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = M,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)
  disp <- model$imputed$dispersion(M)
  plot_dispersion(disp)
}

```

---

plot\_embedding

*Plot Embedding with Improved API*


---

**Description**

Simplified interface for plotting embeddings with automatic caching. Model is the first argument, and color\_by handles metadata/genes automatically.

**Usage**

```

plot_embedding(
  model,
  embedding = NULL,
  color_by = NULL,
  color = NULL,
  projection = "zdb",
  color_limits = NULL,
  palette = c("blue", "lightgrey", "red"),
  randomize = TRUE,
  point_size = 0.3,
  alpha = 0.9,
  raster = FALSE,
  xlab = "Dim 1",
  ylab = "Dim 2",
  title = NULL,
  legend_title = NULL,
  verbose = TRUE
)

```

**Arguments**

|              |                                                                     |
|--------------|---------------------------------------------------------------------|
| model        | GEDI model object (or embedding matrix for backwards compatibility) |
| embedding    | Character ("umap", "pca") or Nx2 matrix                             |
| color_by     | Character: "sample", metadata column, gene name, or NULL            |
| color        | Vector for manual coloring (overrides color_by)                     |
| projection   | Character: "zdb" or "db" for gene expression projection             |
| color_limits | Numeric vector c(low, high) or NULL for auto                        |
| palette      | Character vector of colors for continuous scale                     |
| randomize    | Logical, randomize point order                                      |
| point_size   | Numeric, size of points                                             |
| alpha        | Numeric, transparency (0-1)                                         |
| raster       | Logical, use rasterization for large datasets                       |
| xlab         | Character, x-axis label                                             |
| ylab         | Character, y-axis label                                             |
| title        | Character, plot title                                               |
| legend_title | Character, legend title                                             |
| verbose      | Logical, print computation messages                                 |

**Value**

ggplot2 object

**Examples**

```
if (requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIOobject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)
  plot_embedding(model, embedding = "pca", color_by = "sample")
}
```

plot\_features

*Plot Multiple Features on Embedding***Description**

Efficiently plots multiple gene or pathway features on a 2D embedding using faceting. For standard projections ("zdb", "db", "adb") the values are computed on-the-fly without materialising the full  $J \times N$  matrix. For differential projections ("diffexp", "diffadb") a contrast vector must be supplied.

**Usage**

```
plot_features(
  model,
  features,
  embedding = "umap",
  projection = "zdb",
  contrast = NULL,
  include_0 = FALSE,
  color_limits = "global",
  ncol = NULL,
  randomize = TRUE,
  point_size = 0.2,
  alpha = 0.9,
  title = NULL
)
```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                                           |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| model      | GEDI model object (trained).                                                                                                                                                                                                                                                                                              |
| features   | Character vector of gene/pathway names or integer indices. For "zdb", "db", and "diffexp" these are gene identifiers. For "adb" and "diffadb" these are pathway names (column names of the C matrix supplied at setup).                                                                                                   |
| embedding  | Character specifying embedding type ("umap", "pca") or a custom $N \times 2$ matrix.                                                                                                                                                                                                                                      |
| projection | Character, type of projection to compute. One of "zdb" (shared manifold), "db" (latent factor embedding), "adb" (pathway activity), "diffexp" (differential expression), or "diffadb" (differential pathway activity). "diffexp" and "diffadb" require contrast to be set. "diffexp" optionally accepts include_0 = TRUE. |
| contrast   | Numeric vector of length equal to the number of original H covariates (ncol(model\$priors\$H.rotation). Required when projection is "diffexp" or "diffadb"; ignored otherwise. The vector encodes the linear contrast in the user-facing covariate space.                                                                 |
| include_0  | Logical (default FALSE). When projection = "diffexp", adds the global offset differential effect (diffO) to each gene's projection value. Ignored for all other projection types.                                                                                                                                         |

|              |                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------|
| color_limits | Character ("global" for shared scale, "individual" for per-facet scale) or numeric vector c(low, high). |
| ncol         | Integer, number of columns in facet layout.                                                             |
| randomize    | Logical, whether to randomize point order before plotting.                                              |
| point_size   | Numeric, size of points.                                                                                |
| alpha        | Numeric, transparency of points (0-1).                                                                  |
| title        | Character, plot title (auto-derived from projection type if NULL).                                      |

### Value

ggplot2 object with faceted features.

### Examples

```
if (requireNamespace("SeuratObject", quietly = TRUE) &&
    requireNamespace("uwot", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)
  plot_features(model, c(1, 2), embedding = "pca")
}
# Differential expression projection (requires an H design matrix at setup)
set.seed(42)
J <- 40; N <- 60; n_samples <- 2
M <- Matrix::Matrix(matrix(rpois(J * N, 5), J, N), sparse = TRUE)
rownames(M) <- paste0("Gene_", seq_len(J))
colnames(M) <- paste0("Cell_", seq_len(N))
samples <- factor(rep(paste0("S", seq_len(n_samples))), each = N / n_samples)
H <- matrix(c(0, 1), nrow = 1,
             dimnames = list("condition", paste0("S", seq_len(n_samples))))
model <- CreateGEDIObject(Samples = samples, M = M, K = 3, H = H, verbose = 0)
model$train(iterations = 10)
contrast <- c(condition = 1) # length == ncol(model$priors$H.rotation) == 1
p <- plot_features(model, c("Gene_1", "Gene_2"),
                  embedding = "pca",
                  projection = "diffexp",
                  contrast = contrast)
```

---

|                    |                                    |
|--------------------|------------------------------------|
| plot_feature_ratio | <i>Plot Two-Feature Comparison</i> |
|--------------------|------------------------------------|

---

## Description

Compares two features by computing their difference or correlation in the projected space. Mathematically grounded for GEDI's log-space representation.

## Usage

```
plot_feature_ratio(
  model,
  gene1,
  gene2,
  comparison = "difference",
  embedding = "umap",
  projection = "zdb",
  color_limits = NULL,
  randomize = TRUE,
  point_size = 0.3,
  alpha = 0.9,
  title = NULL
)
```

## Arguments

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| model        | GEDI model object                                                            |
| gene1        | Character or integer, first gene name or index                               |
| gene2        | Character or integer, second gene name or index                              |
| comparison   | Character, type of comparison ("difference" or "correlation")                |
| embedding    | Character specifying embedding type ("umap", "pca") or a custom N x 2 matrix |
| projection   | Character, type of projection ("zdb" or "db")                                |
| color_limits | Numeric vector c(low, high) or NULL for auto-compute                         |
| randomize    | Logical, whether to randomize point order                                    |
| point_size   | Numeric, size of points                                                      |
| alpha        | Numeric, transparency of points                                              |
| title        | Character, plot title                                                        |

## Details

For comparison = "difference": Computes  $(Z[\text{gene1},] - Z[\text{gene2},]) * D * B$ , equivalent to  $ZDB[\text{gene1},] - ZDB[\text{gene2},]$ . In log-space, this represents  $\log(\text{gene1}/\text{gene2})$  in the original count space. Positive values indicate  $\text{gene1} > \text{gene2}$ , negative indicates  $\text{gene2} > \text{gene1}$ .

**Value**

ggplot2 object

**Examples**

```
if (requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIOObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)
  gene1 <- rownames(pbmc_small)[1]
  gene2 <- rownames(pbmc_small)[2]
  plot_feature_ratio(model, gene1, gene2, comparison = "difference",
                    embedding = "pca")
}
```

---

|                   |                                                 |
|-------------------|-------------------------------------------------|
| plot_vector_field | <i>Plot Vector Field from Dynamics Analysis</i> |
|-------------------|-------------------------------------------------|

---

**Description**

Visualizes vector fields showing cell state transitions. Uses binned aggregation for cleaner visualization without overplotting.

**Usage**

```
plot_vector_field(
  dynamics_svd,
  color = NULL,
  alpha = 1,
  n_bins = 50,
  min_per_bin = 10,
  randomize = TRUE,
  arrow_size = 0.5,
  arrow_length = 0.15,
  arrow_color = "black",
  dims = c(1, 2),
  xlab = NULL,
  ylab = NULL,
  title = NULL
)
```

**Arguments**

|              |                                                        |
|--------------|--------------------------------------------------------|
| dynamics_svd | Result from model\$dynamics\$vector_field() or similar |
| color        | Vector of length N for coloring arrows, or NULL        |
| alpha        | Vector of length N or scalar for arrow transparency    |
| n_bins       | Integer, number of bins per dimension for aggregation  |
| min_per_bin  | Integer, minimum observations required per bin         |
| randomize    | Logical, whether to randomize data order               |
| arrow_size   | Numeric, size of arrow lines                           |
| arrow_length | Numeric, length of arrow heads (in cm)                 |
| arrow_color  | Character, color for arrows (if color is NULL)         |
| dims         | Integer vector of length 2, which dimensions to plot   |
| xlab         | Character, x-axis label                                |
| ylab         | Character, y-axis label                                |
| title        | Character, plot title                                  |

**Value**

ggplot2 object

**Examples**

```
# Build a tiny multi-sample fixture with a sample-level prior H so that
# model$dynamics$vector_field() is available.
set.seed(1)
n_genes <- 80; n_cells <- 60; n_samples <- 3
M <- Matrix::Matrix(
  matrix(stats::rpois(n_genes * n_cells, 5), n_genes, n_cells),
  sparse = TRUE
)
rownames(M) <- paste0("G", seq_len(n_genes))
colnames(M) <- paste0("C", seq_len(n_cells))
samples <- factor(rep(paste0("S", seq_len(n_samples))),
  each = n_cells / n_samples)
H <- matrix(c(1, 0, 0, 0, 1, 0), nrow = 2, byrow = TRUE)
colnames(H) <- paste0("S", seq_len(n_samples))
rownames(H) <- c("cond_a", "cond_b")

model <- CreateGEDIOObject(Samples = samples, M = M, K = 3, H = H,
  verbose = 0)
model$train(iterations = 5)
vf <- model$dynamics$vector_field(start.cond = c(1, 0),
  end.cond = c(0, 1))

plot_vector_field(vf)
```

read\_h5

*Read 10X Genomics H5 file***Description**

Reads a 10X Genomics HDF5 file (CellRanger v2/v3 format) and converts it to a sparse dgCMatrx suitable for use with gedi R6 object.

**Usage**

```
read_h5(
  file_path,
  feature_format = "gene_name",
  unique.features = TRUE,
  verbose = FALSE
)
```

**Arguments**

`file_path` Character. Path to the 10X H5 file.

`feature_format` Character. Which feature identifier to use for gene names. Options: "gene\_name" (default, uses feature names) or "gene\_ids" (uses feature IDs).

`unique.features` Logical. Make feature names unique. Default TRUE.

`verbose` Logical. Print progress messages. Default FALSE.

**Value**

Sparse matrix (dgCMatrx) with genes as rows and cells as columns.

**Examples**

```
# read_h5() expects a 10x Genomics filtered_feature_bc_matrix.h5 file.
# The example body runs only when such a file is on disk and hdf5r is
# installed; otherwise it is silently skipped.
h5_file <- "filtered_feature_bc_matrix.h5"
if (file.exists(h5_file) &&
    requireNamespace("hdf5r", quietly = TRUE)) {
  expr_matrix <- read_h5(h5_file)
  expr_matrix_ids <- read_h5(h5_file, feature_format = "gene_ids")
}
```

read\_h5ad

*Read H5AD file and convert to sparse matrix***Description**

Reads an H5AD file (AnnData format) and extracts the expression matrix as a sparse dgCMatrx suitable for use with gedi R6 object.

**Usage**

```
read_h5ad(
  file_path,
  layer = NULL,
  use_raw = FALSE,
  transpose = TRUE,
  return_metadata = FALSE,
  feature_format = "gene_name",
  verbose = FALSE
)
```

**Arguments**

|                 |                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file_path       | Character. Path to the H5AD file.                                                                                                                                                                            |
| layer           | Character. The layer to extract from the H5AD file. Default is NULL, which reads from X (the main expression matrix). Common alternatives include "counts", "data", "scaled", etc.                           |
| use_raw         | Logical. If TRUE, reads from the raw.X slot instead of X. Default is FALSE.                                                                                                                                  |
| transpose       | Logical. If TRUE, transposes the matrix so genes are rows and cells are columns (gedi format). Default is TRUE.                                                                                              |
| return_metadata | Logical. If TRUE, returns a list with the expression matrix, cell metadata (obs), and gene metadata (var). If FALSE, returns only the expression matrix. Default is FALSE.                                   |
| feature_format  | Character. Which feature identifier to use for gene names when return_metadata = FALSE. Options: "gene_name" (default, uses var rownames) or "gene_ids" (uses var\$gene_ids column). Default is "gene_name". |
| verbose         | Logical. If TRUE, prints detailed progress messages. Default is FALSE.                                                                                                                                       |

**Details**

This function reads H5AD files, which are the standard format for AnnData objects in Python. Compatible with gedi R6 class for seamless integration.

**Value**

If `return_metadata = FALSE`, returns a sparse matrix (`dgCMatrix`) with genes as rows and cells as columns. If `return_metadata = TRUE`, returns a list with components:

- `X`: sparse expression matrix (genes x cells)
- `obs`: `data.frame` of cell metadata
- `var`: `data.frame` of gene metadata

**Examples**

```
# Round-trip: write a tiny H5AD via write_h5ad(), then read it back.
if (requireNamespace("hdf5r", quietly = TRUE) &&
    requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M        = pbmc_small@assays$RNA@counts,
    K        = 3,
    verbose  = 0
  )
  model$train(iterations = 5)

  tmp <- tempfile(fileext = ".h5ad")
  write_h5ad(model, tmp)
  expr_matrix <- read_h5ad(tmp)
  unlink(tmp)
}
```

---

seurat\_to\_gedi

---

*Convert Seurat Object to GEDI Model*


---

**Description**

Extracts count data from a Seurat object and creates a GEDI model. Automatically validates that the data contains raw counts (not normalized). Handles both Seurat v4 and v5, including split layers in v5.

**Usage**

```
seurat_to_gedi(
  seurat_object,
  assay = "RNA",
  slot = "counts",
  sample_column = "orig.ident",
  subset_samples = NULL,
  K = 10,
```

```

mode = "B12",
C = NULL,
H = NULL,
validate_counts = TRUE,
use_variable_features = TRUE,
verbose = TRUE,
...
)

```

## Arguments

|                       |                                                                                                                                                                                               |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| seurat_object         | Seurat object                                                                                                                                                                                 |
| assay                 | Character, which assay to use (default: "RNA")                                                                                                                                                |
| slot                  | Character, which slot/layer to extract (default: "counts"). For Seurat v5 with split layers (e.g., counts.CTRL, counts.STIM), this will automatically detect and combine all matching layers. |
| sample_column         | Character, column name in meta.data for sample labels (default: "orig.ident")                                                                                                                 |
| subset_samples        | Character vector, subset to specific samples (default: NULL = all)                                                                                                                            |
| K                     | Integer, number of latent factors (default: 10)                                                                                                                                               |
| mode                  | Character, normalization mode: "B12" or "Bsphere" (default: "B12")                                                                                                                            |
| C                     | Gene-level prior matrix (genes x pathways) (default: NULL)                                                                                                                                    |
| H                     | Sample-level covariate matrix (covariates x samples) (default: NULL)                                                                                                                          |
| validate_counts       | Logical, whether to validate data appears to be counts (default: TRUE)                                                                                                                        |
| use_variable_features | Logical, whether to subset to highly variable features (default: TRUE). If TRUE, uses genes from VariableFeatures(seurat_object).                                                             |
| verbose               | Logical, whether to print progress messages (default: TRUE)                                                                                                                                   |
| ...                   | Additional arguments passed to CreateGEDIOObject()                                                                                                                                            |

## Value

GEDIR6 object

## Examples

```

library(Seurat)

# Load example data
pbmc <- SeuratObject::pbmc_small

# Basic usage
gedi_model <- seurat_to_gedi(
  seurat_object = pbmc,
  sample_column = "orig.ident",
  K = 15
)

```

```

)

# Train the model
gedi_model$train(iterations = 10)

```

---

write\_h5ad

---

Write GEDI model to H5AD file

---

## Description

Exports a trained GEDI model to H5AD (AnnData) format for interoperability with Python tools like scanpy. The file contains expression data, embeddings, metadata, and GEDI-specific parameters.

## Usage

```

write_h5ad(
  model,
  file_path,
  X_slot = c("imputed", "projection", "original"),
  M = NULL,
  include_embeddings = TRUE,
  include_raw = FALSE,
  compression = 6,
  verbose = TRUE
)

```

## Arguments

|                    |                                                                                                                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| model              | GEDI R6 object (must be trained)                                                                                                                                                                                                                                                                 |
| file_path          | Character. Path where the H5AD file should be written.                                                                                                                                                                                                                                           |
| X_slot             | Character. Which expression data to save in the main X slot: <ul style="list-style-type: none"> <li>"imputed": Imputed expression (default, requires M matrix)</li> <li>"projection": ZDB projection (always available)</li> <li>"original": Original M matrix (requires M parameter)</li> </ul> |
| M                  | Optional. Original count matrix to save when X_slot="original" or include_raw=TRUE. Must match the dimensions used during model training.                                                                                                                                                        |
| include_embeddings | Logical. Include PCA/UMAP in obsm if cached. Default TRUE.                                                                                                                                                                                                                                       |
| include_raw        | Logical. If TRUE, saves original M in raw.X (requires M parameter). Default FALSE.                                                                                                                                                                                                               |
| compression        | Integer. Gzip compression level (0-9). Default 6.                                                                                                                                                                                                                                                |
| verbose            | Logical. Print progress messages. Default TRUE.                                                                                                                                                                                                                                                  |

## Details

The H5AD file structure contains:

- X: Main expression matrix (based on X\_slot parameter)
- obs: Cell metadata (sample IDs, colData)
- var: Gene metadata (gene IDs)
- obsm: Cell embeddings (X\_gedi, X\_pca, X\_umap)
- varm: Gene loadings (gedi\_Z, gedi\_Q\_mean)
- uns: Model parameters and metadata
- raw.X: Original counts (if include\_raw=TRUE)

The function handles the technical details of HDF5/AnnData compatibility:

- Writes sparse matrices in CSR format
- Transposes dense matrices for Python's row-major layout
- Creates scalar string attributes (not arrays) for AnnData compatibility
- Validates M matrix identity using fingerprints

## Value

Invisibly returns the file path

## Examples

```
if (requireNamespace("hdf5r", quietly = TRUE) &&
    requireNamespace("SeuratObject", quietly = TRUE)) {
  pbmc_small <- SeuratObject::pbmc_small
  model <- CreateGEDIObject(
    Samples = pbmc_small@meta.data$orig.ident,
    M       = pbmc_small@assays$RNA@counts,
    K       = 3,
    verbose = 0
  )
  model$train(iterations = 5)

  # Write to a temporary file (CRAN policy: never write to the user's
  # working directory in examples).
  tmp <- tempfile(fileext = ".h5ad")
  write_h5ad(model, tmp)
  write_h5ad(model, tmp, X_slot = "projection")
  unlink(tmp)
}
```

# Index

`check_optional_dependencies`, [2](#)  
`CreateGEDIOObject`, [3](#)  
  
`gedi_to_seurat`, [5](#)  
  
`install_optional_dependencies`, [6](#)  
  
`list_h5_structure`, [7](#)  
  
`plot_convergence`, [8](#)  
`plot_dispersion`, [9](#)  
`plot_embedding`, [10](#)  
`plot_feature_ratio`, [14](#)  
`plot_features`, [12](#)  
`plot_vector_field`, [15](#)  
  
`read_h5`, [17](#)  
`read_h5ad`, [18](#)  
  
`seurat_to_gedi`, [19](#)  
  
`write_h5ad`, [21](#)