

# Package ‘tomoda’

February 2, 2026

**Title** Tomo-seq data analysis

**Version** 1.21.0

**Description** This package provides many easy-to-use methods to analyze and visualize tomo-seq data. The tomo-seq technique is based on cryosectioning of tissue and performing RNA-seq on consecutive sections. (Reference: Kruse F, Junker JP, van Oudenaarden A, Bakkers J. Tomo-seq: A method to obtain genome-wide expression data with spatial resolution. *Methods Cell Biol.* 2016;135:299-307. doi:10.1016/bs.mcb.2016.01.006)

The main purpose of the package is to find zones with similar transcriptional profiles and spatially expressed genes in a tomo-seq sample. Several visualization functions are available to create easy-to-modify plots.

**Depends** R (>= 4.0.0)

**Imports** methods, stats, grDevices, reshape2, Rtsne, umap, RColorBrewer, ggplot2, ggrepel, SummarizedExperiment

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Suggests** knitr, rmarkdown, BiocStyle, testthat

**URL** <https://github.com/liuwd15/tomoda>

**BugReports** <https://github.com/liuwd15/tomoda/issues>

**biocViews** GeneExpression, Sequencing, RNASeq, Transcriptomics, Spatial, Clustering, Visualization

**VignetteBuilder** knitr

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

**git\_branch** devel

**git\_last\_commit** 2682c14

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2026-02-01

**Author** Wenda Liu [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5124-9338>>)

**Maintainer** Wenda Liu <liuwd15@tsinghua.org.cn>

## Contents

corHeatmap	2
createTomo	3
embedPlot	5
expHeatmap	6
findPeak	7
findPeakGene	7
geneCorHeatmap	9
geneEmbedPlot	10
hierarchClust	11
kmeansClust	12
linePlot	13
normalizeTomo	14
runPCA	15
runTSNE	16
runUMAP	17
scaleTomo	18
tomoMatrix	18
tomoSummarizedExperiment	19
zh.data	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

corHeatmap	<i>Correlation heatmap of sections</i>
------------	--

---

### Description

Heatmap pf correlation coefficients between any two sections in a SummarizedExperiment object.

### Usage

```
corHeatmap(object, matrix = "scaled", max.cor = 0.5, cor.method = "pearson")
```

### Arguments

object	A SummarizedExperiment object.
matrix	Character, must be one of "count", "normalized", or "scaled".
max.cor	Numeric, correlation coefficients bigger than max.cor are set to max.cor. It is used to clearly show small correlation coefficients.
cor.method	Character, the method to calculate correlation coefficients. must be one of "pearson", "kendall", or "spearman".

**Value**

A ggplot object.

**Examples**

```
data(zh.data)
zh <- createTomo(zh.data)
corHeatmap(zh)

# Use Spearman correlation coefficients.
corHeatmap(zh, cor.method='spearman')

# Set max correlation coefficients to 0.3.
corHeatmap(zh, max.cor=0.3)
```

---

**createTomo***Create an object representing tomo-seq data*

---

**Description**

This is a generic function to create an object representing tomo-seq data. The input object can either be a `matrix` or a `SummarizeExperiment`.

**Usage**

```
createTomo(object, ...)

## S4 method for signature 'SummarizedExperiment'
createTomo(
  object,
  min.section = 3,
  normalize = TRUE,
  normalize.method = "median",
  scale = TRUE
)

## S4 method for signature 'matrix'
createTomo(
  object,
  matrix.normalized = NULL,
  min.section = 3,
  normalize = TRUE,
  normalize.method = "median",
  scale = TRUE
)

## S4 method for signature 'missing'
```

```
createTomo(
  matrix.normalized = NULL,
  min.section = 3,
  normalize = TRUE,
  normalize.method = "median",
  scale = TRUE,
  ...
)
```

## Arguments

object	Either a raw read count matrix or a SummarizedExperiment object.
...	Additional parameters to pass to S4 methods.
min.section	Integer. Genes expressed in less than min.section sections will be filtered out.
normalize	Logical, whether to perform normalization when creating the object. Default is TRUE.
normalize.method	Character, must be one of "median", or "cpm".
scale	Logical, whether to perform scaling when creating the object. Default is TRUE.
matrix.normalized	(Optional) A numeric matrix of normalized read count.

## Details

This is the generic function to create a SummarizedExperiment object for representing tomo-seq data. Either matrix or SummarizedExperiment object can be used for input.

When using matrix for input, at least one of raw read count matrix and normalized read count matrix (like FPKM and TPM) must be used for input. If normalized matrix is available, input it with argument matrix.normalized. Matrices should have genes as rows and sections as columns. Columns should be sorted according to the order of sections.

When using SummarizedExperiment object for input, it must contain at least one of 'count' assay and 'normalized' assay. Besides, the row data and column data of the input object will be retained in the output object.

By default, all library sizes are normalized to the median library size across sections. Set normalize.method = "cpm" will make library sizes normalized to 1 million counts. Scaling and centering is performed for all genes across sections.

## Value

A SummarizedExperiment object. Raw read count matrix, normalized read count matrix and scaled read count matrix are saved in 'count', 'normalized' and 'scale' assays of the object.

## See Also

- [tomoMatrix](#) : creating an object from matrix.
- [tomoSummarizedExperiment](#) : creating an object from SummarizedExperiment.

- `normalizeTomo` : normalization.
- `scaleTomo` : scaling.
- `SummarizedExperiment-class` : operations on `SummarizedExperiment`.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

data(zh.data)
se <- SummarizedExperiment::SummarizedExperiment(assays=list(count=zh.data))
zh <- createTomo(se)
```

---

embedPlot

*Embedding plot for sections*

---

## Description

Scatter plot for sections with two-dimensional embeddings in a `SummarizedExperiment` object. Each point stands for a section.

## Usage

```
embedPlot(object, group = "section", method = "TSNE")
```

## Arguments

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>group</code>	Character, a variable in slot <code>meta</code> defining the groups of sections. Sections in the same group have same colors.
<code>method</code>	Character, the embeddings for scatter plot. Must be one of "TSNE", "UMAP", or "PCA".

## Value

A `ggplot` object.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)
zh <- runTSNE(zh)
# Plot TSNE embeddings.
embedPlot(zh)

# Plot UMAP embeddings.
zh <- runUMAP(zh)
embedPlot(zh, method="UMAP")
```

```
# Color sections by kmeans cluster labels.
zh <- kmeansClust(zh, 3)
embedPlot(zh, group="kmeans_cluster")
```

---

expHeatmap

*Expression heatmap*

---

## Description

Heatmap for gene expression across sections in a `SummarizedExperiment` object.

## Usage

```
expHeatmap(object, genes, matrix = "scaled", size = 5)
```

## Arguments

object	A <code>SummarizedExperiment</code> object.
genes	A vector of character, the name of genes to plot heatmap.
matrix	Character, must be one of "count", "normalized", or "scaled".
size	Character, the size of gene names. Set it to 0 if you do not want to show gene names.

## Value

A `ggplot` object.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

# Plot some genes.
expHeatmap(zh,
c("ENSDARG00000002131", "ENSDARG00000003061", "ENSDARG00000076075", "ENSDARG00000076850"))

# Plot peak genes.
peak_genes <- findPeakGene(zh)
expHeatmap(zh, peak_genes$gene)

# Remove gene names if too many genes are in the heatmap.
expHeatmap(zh, peak_genes$gene, size=0)
```

---

**findPeak***Find peak in a vector*

---

## Description

Find the position of peak in a vector.

## Usage

```
findPeak(x, threshold = 1, length = 4)
```

## Arguments

<code>x</code>	A numeric vector.
<code>threshold</code>	Integer, only values bigger than <code>threshold</code> are recognized as part of the peak.
<code>length</code>	Integer, minimum length of consecutive values bigger than <code>threshold</code> are recognized as a peak.

## Value

A numeric vector. The first element is the start index and the second element is the end index of the peak. If multiple peaks exist, only output the start and end index of the one with maximum length. If no peak exist, return `c(0, 0)`.

## Examples

```
# return c(3, 10)
findPeak(c(0:5, 5:0), threshold=1, length=4)

# Most likely return c(0, 0)
findPeak(rnorm(10), threshold=3, length=3)
```

---

**findPeakGene***Find peak genes*

---

## Description

Find peak genes (spatially upregulated genes) in a `SummarizedExperiment` object.

## Usage

```
findPeakGene(
  object,
  threshold = 1,
  length = 4,
  matrix = "scaled",
  nperm = 1e+05,
  method = "BH"
)
```

## Arguments

object	A SummarizedExperiment object.
threshold	Integer, only scaled read counts bigger than threshold are recognized as part of the peak.
length	Integer, scaled read counts bigger than threshold in minimum length of consecutive sections are recognized as a peak.
matrix	Character, must be one of "count", "normalized", or "scaled".
nperm	Integer, number of random permutations to calculate p values. Set it to 0 if you are not interested in p values.
method	Character, the method to adjust p values for multiple comparisons, must be one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

## Details

Peak genes are selected based on scaled read counts. As scaled read counts are Z scores, suggested threshold are [1, 3]. Smaller threshold and length makes the function detect more peak genes, and vice versa. P values are calculated by approximate permutation tests. For a given threshold and length, the scaled read counts of each gene is randomly permuted for nperm times. The p value is defined as the ratio of permutations containing peaks. In order to speed up permutation process, genes whose expression exceeds threshold in same number of sections have same p values. To be specific, only one of these genes will be used to calculate a p value by permutation, and other genes are assigned this p value.

## Value

A data.frame with peak genes as rows. It has following columns:

- gene : Character, peak gene names.
- start : Numeric, the start index of peak.
- end : Numeric, the end index of peak.
- center : Numeric, the middle index of peak. If the length of the peak is even, center is defined as the left-middle index.
- p : Numeric, p values.
- p.adj : Numeric, adjusted p values.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)
peak_genes <- findPeakGene(zh)
head(peak_genes)

# Increase threshold so that less peak genes will be found.
peak_genes <- findPeakGene(zh, threshold=1.5)

# Increase peak length so that less peak genes will be found.
peak_genes <- findPeakGene(zh, length=5)

# Set nperm to 0 so that p values will not be calculated. This will save running time.
peak_genes <- findPeakGene(zh, nperm=0)
```

---

geneCorHeatmap

*Correlation heatmap of genes*

---

## Description

Heatmap of correlation coefficients between any two queried genes in a `SummarizedExperiment` object.

## Usage

```
geneCorHeatmap(
  object,
  gene.df,
  group = "center",
  matrix = "scaled",
  size = 5,
  cor.method = "pearson"
)
```

## Arguments

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>gene.df</code>	Data.frame. The first column must be a vector of gene names, and has the name "gene". Additional columns in <code>gene.df</code> can be used to set the colors of genes.
<code>group</code>	Character, a column name in <code>gene.df</code> defining the groups of genes. Genes in the same group have same colors on the side bar.
<code>matrix</code>	Character, must be one of "count", "normalized", or "scaled".
<code>size</code>	Numeric, the size of gene names. Set it to 0 if you do not want to show gene names.
<code>cor.method</code>	Character, the method to calculate correlation coefficients. must be one of "pearson", "kendall", or "spearman".

## Details

This method can create a pure heatmap or a heatmap with side bar. If you prefer a pure heatmap, input a gene.df with a single column of gene names. However, you may want to show additional information of genes with a side bar, and the grouping information should be saved as additional column(s) of gene.df, and declared as group. By default, you can use the output by `findPeakGene` as input gene.df. Peak genes will be grouped by their centers on the side bar.

## Value

A ggplot object.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

# Correlation heatmap for all peak genes.
peak_genes <- findPeakGene(zh)
geneCorHeatmap(zh, peak_genes)

# Use Spearman correlation coefficients.
geneCorHeatmap(zh, peak_genes, cor.method="spearman")

# Group genes by peak start.
geneCorHeatmap(zh, peak_genes, group="start")

# Plot without side bar.
geneCorHeatmap(zh, data.frame(
  gene=c("ENSDARG00000002131", "ENSDARG00000003061", "ENSDARG00000076075", "ENSDARG00000076850")))
```

geneEmbedPlot

*Embedding plot for genes*

## Description

Scatter plot for genes with two-dimenstional embeddings in a `SummarizedExperiment` object. Each point stands for a gene.

## Usage

```
geneEmbedPlot(object, gene.df, group = "center", method = "TSNE")
```

## Arguments

object	A <code>SummarizedExperiment</code> object.
gene.df	Data.frame. The first column must be a vector of gene names, and has the name "gene". Additional columns in gene.df can be used to set the colors of genes.

group	Character, a column name in gene.df defining the groups of genes. Genes in the same group have same colors.
method	Character, the embeddings for scatter plot. Must be one of "TSNE", "UMAP", or "PCA".

### Value

A ggplot object.

### Examples

```
data(zh.data)
zh <- createTomo(zh.data)
peak_genes <- findPeakGene(zh)
zh <- runTSNE(zh, peak_genes$gene)
# Color genes by peak centers.
geneEmbedPlot(zh, peak_genes)

# Color genes by peak starts.
geneEmbedPlot(zh, peak_genes, group="start")

# Do not color genes.
geneEmbedPlot(zh, peak_genes["gene"])
```

---

## hierarchClust

*Hierarchical clustering across sections*

---

### Description

Performs hierarchical clustering across sections in a SummarizedExperiment object.

### Usage

```
hierarchClust(
  object,
  matrix = "normalized",
  measure = "euclidean",
  p = 2,
  agglomeration = "complete"
)
```

### Arguments

object	A SummarizedExperiment object.
matrix	Character, must be one of "count", "normalized", or "scaled".
measure	Character, must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".

**p** Numeric, the power of the Minkowski distance.

**agglomeration** Character, must be one of "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".

### Value

A `hclust` object.

### See Also

`dist` for measuring distance and `hclust` for performing hierarchical clustering on a matrix.

### Examples

```
data(zh.data)
zh <- createTomo(zh.data)
hclust_zh <- hierarchClust(zh)
plot(hclust_zh)

# Use other agglomeration method
hclust_zh <- hierarchClust(zh, agglomeration="average")

# (Not recommended) Use scaled read counts to calculate distance
zh <- scaleTomo(zh)
hclust_zh <- hierarchClust(zh, matrix="scaled")
```

## kmeansClust

*K-Means clustering across sections*

### Description

Performs K-Means clustering across sections in a `SummarizedExperiment` object.

### Usage

```
kmeansClust(object, centers, matrix = "normalized", ...)
```

### Arguments

**object** A `SummarizedExperiment` object.

**centers** Integer, number of clusters, namely  $k$ .

**matrix** Character, must be one of "count", "normalized", or "scaled".

**...** other parameters passed to `kmeans`.

### Value

A `SummarizedExperiment` object. The obtained cluster labels are saved in slot `meta`.

**See Also**

[kmeans](#) for performing K-Means clustering on a matrix.

**Examples**

```
data(zh.data)
zh <- createTomo(zh.data)
zh <- kmeansClust(zh, 3)

# Use scaled read counts to calculate distance
zh <- scaleTomo(zh)
zh <- kmeansClust(zh, 3, matrix="scaled")
```

---

**linePlot***Line plot for expression traces*

---

**Description**

Plot expression traces for genes across sections in a `SummarizedExperiment` object.

**Usage**

```
linePlot(object, genes, matrix = "normalized", facet = FALSE, span = 0.3)
```

**Arguments**

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>genes</code>	A character vector of gene names for plotting expression traces.
<code>matrix</code>	Character, must be one of "count", "normalized", or "scaled".
<code>facet</code>	Logical. Plot the expression trace of each gene in a facet if it is TRUE.
<code>span</code>	Numeric, the amount of smoothing for the default loess smoother. Smaller numbers produce wigglier lines, larger numbers produce smoother lines. Set it to 0 for non-smoothing lines.

**Value**

A `ggplot` object.

**See Also**

`geom_smooth` for plotting smooth lines, `facet_wrap` for faceting genes.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)
linePlot(zh,
  c("ENSDARG0000002131", "ENSDARG0000003061", "ENSDARG00000076075", "ENSDARG00000076850"))

# Do not smooth lines.
linePlot(zh,
  c("ENSDARG0000002131", "ENSDARG0000003061", "ENSDARG00000076075", "ENSDARG00000076850"), span=0)

# Plot genes in different facets.
linePlot(zh,
  c("ENSDARG0000002131", "ENSDARG0000003061", "ENSDARG00000076075", "ENSDARG00000076850"),
  facet=TRUE)
```

normalizeTomo

*Normalize data*

## Description

Normalize the raw read count in a `SummarizedExperiment` object.

## Usage

```
normalizeTomo(object, method = "median")
```

## Arguments

object	A <code>SummarizedExperiment</code> object.
method	Character, must be one of "median", or "cpm".

## Details

This function should be run for `SummarizedExperiment` object created from raw read count matrix. If the `SummarizedExperiment` object already has a normalized count matrix. The function simply return the original object. Library sizes of all sections are normalized to the median library size (method='median') or one million (method='cpm').

## Value

A `SummarizedExperiment` object with normalized read count matrix saved in assay 'normalized'.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data, normalize=FALSE)
zh <- normalizeTomo(zh)
```

---

runPCA	<i>Perform PCA</i>
--------	--------------------

---

## Description

Perform PCA on sections or genes in a `SummarizedExperiment` object for dimensionality reduction.

## Usage

```
runPCA(object, genes = NA, matrix = "auto", scree = FALSE, ...)
```

## Arguments

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>genes</code>	NA or a vector of character. Perform PCA on sections if it is NA, or on given genes if it is a vector of gene names.
<code>matrix</code>	Character, must be one of "auto", "count", "normalized", or "scaled". If "auto", normalized matrix is used for sections and scaled matrix is used for genes.
<code>scree</code>	Logical, plot the scree plot for PCs if it is TRUE.
...	Other parameters passed to <code>prcomp</code> .

## Value

A `SummarizedExperiment` object. The PC embeddings are saved in slot `meta` if PCA is performed on sections, or saved in slot `gene_embedding` if PCA is performed on genes.

## See Also

[prcomp](#) for performing PCA on a matrix.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

# Perform PCA on sections.
zh <- runPCA(zh)

# Plot the scree plot.
zh <- runPCA(zh, scree=TRUE)

# Perform PCA on some genes.
zh <- runPCA(zh, genes=rownames(zh)[1:100])
```

---

runTSNE*Perform TSNE*

---

## Description

Perform TSNE on sections or genes in a `SummarizedExperiment` object for dimensionality reduction.

## Usage

```
runTSNE(object, genes = NA, matrix = "auto", perplexity = NA, ...)
```

## Arguments

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>genes</code>	NA or a vector of character. Perform TSNE on sections if it is NA, or on given genes if it is a vector of gene names.
<code>matrix</code>	Character, must be one of "auto", "count", "normalized", or "scaled". If "auto", normalized matrix is used for sections and scaled matrix is used for genes.
<code>perplexity</code>	Numeric, perplexity parameter for <code>Rtsne</code> (default: 0.25 *(number of observations - 1)).
<code>...</code>	Other parameters passed to <code>Rtsne</code> .

## Value

A `SummarizedExperiment` object. The TSNE embeddings are saved in slot `meta` if TSNE is performed on sections, or saved in slot `gene_embedding` if TSNE is performed on genes.

## See Also

[Rtsne](#) for performing TSNE on a matrix.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

# Perform TSNE on sections.
zh <- runTSNE(zh)

# Perform TSNE on sections with other perplexity.
zh <- runTSNE(zh, perplexity=10)

# Perform TSNE on some genes.
zh <- runTSNE(zh, genes=rownames(zh)[1:100])
```

---

**runUMAP***Perform UMAP*

---

## Description

Perform UMAP on sections or genes in a `SummarizedExperiment` object for dimensionality reduction.

## Usage

```
runUMAP(object, genes = NA, matrix = "auto", ...)
```

## Arguments

<code>object</code>	A <code>SummarizedExperiment</code> object.
<code>genes</code>	NA or a vector of character. Perform UMAP on sections if it is NA, or on given genes if it is a vector of gene names.
<code>matrix</code>	Character, must be one of "auto", "count", "normalized", or "scaled". If "auto", normalized matrix is used for sections and scaled matrix is used for genes.
...	Other parameters passed to <code>umap</code> .

## Value

A `SummarizedExperiment` object. The UMAP embeddings are saved in slot `meta` if UMAP is performed on sections, or saved in slot `gene_embedding` if UMAP is performed on genes.

## See Also

[umap](#) for performing UMAP on a matrix.

## Examples

```
data(zh.data)
zh <- createTomo(zh.data)

# Perform UMAP on sections.
zh <- runUMAP(zh)

# Perform UMAP on some genes.
zh <- runUMAP(zh, genes=rownames(zh)[1:100])
```

---

scaleTomo	<i>Scale data</i>
-----------	-------------------

---

### Description

Scale the normalized read count in a `SummarizedExperiment` object.

### Usage

```
scaleTomo(object)
```

### Arguments

object	A <code>SummarizedExperiment</code> object.
--------	---

### Details

This function should be run for `SummarizedExperiment` object with normalized read count matrix. The normalized read counts of each gene are subjected to Z score transformation across sections.

### Value

A `SummarizedExperiment` object with scaled read count matrix saved in assay 'scaled'.

### Examples

```
data(zh.data)
zh <- createTomo(zh.data, scale=FALSE)
zh <- scaleTomo(zh)
```

---

tomoMatrix	<i>Create an object from matrix</i>
------------	-------------------------------------

---

### Description

`tomoMatrix` creates an object from raw read count matrix or normalized read count matrix.

### Usage

```
tomoMatrix(
  matrix.count = NULL,
  matrix.normalized = NULL,
  min.section = 3,
  normalize = TRUE,
  normalize.method = "median",
  scale = TRUE
)
```

## Arguments

`matrix.count` A numeric matrix or matrix-like data structure that can be converted to matrix, with genes with rows, sections as columns and values as raw read counts. Columns should be sorted according to section numbers.

`matrix.normalized` A numeric matrix or matrix-like data structure that can be converted to matrix, with genes as rows, sections as columns and values as normalized read counts. Columns should be sorted according to order of sections.

`min.section` Integer. Genes expressed in less than `min.section` sections will be filtered out.

`normalize` Logical, whether to perform normalization when creating the object. Default is `TRUE`.

`normalize.method` Character, must be one of `"median"`, or `"cpm"`.

`scale` Logical, whether to perform scaling when creating the object. Default is `TRUE`.

## Value

A `SummarizedExperiment` object

## See Also

`createTomo` for the generic function.

## Examples

```
data(zh.data)
zh <- tomoMatrix(zh.data)
```

### tomoSummarizedExperiment

*Create an object from SummarizedExperiment*

## Description

`tomoSummarizedExperiment` creates an object from a `SummarizedExperiment` object.

## Usage

```
tomoSummarizedExperiment(
  se,
  min.section = 3,
  normalize = TRUE,
  normalize.method = "median",
  scale = TRUE
)
```

## Arguments

se	A SummarizedExperiment object, it must contain at least one of 'count' assay and 'normalized' assay.
min.section	Integer. Genes expressed in less than min.section sections will be filtered out.
normalize	Logical, whether to perform normalization when creating the object. Default is TRUE.
normalize.method	Character, must be one of "median", or "cpm".
scale	Logical, whether to perform scaling when creating the object. Default is TRUE.

## Value

A SummarizedExperiment object

## See Also

[createTomo](#) for the generic function.

## Examples

```
data(zh.data)
se <- SummarizedExperiment::SummarizedExperiment(assays=list(count=zh.data))
zh <- tomoSummarizedExperiment(se)
```

zh.data

*A raw read count matrix of zebrafish injured heart.*

## Description

A dataset containing gene expression across 40 sections of zebrafish heart generated with Tomo-seq. The zebrafish heart is 3 days post cryoinjury (3 dpi).

## Usage

```
data(zh.data)
```

## Format

A numeric matrix with 16495 genes as rows and 40 section as columns. Its row names are gene names and column names are section names.

## Source

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74652>

**References**

Wu CC, Kruse F, Vasudevarao MD, et al. Spatially Resolved Genome-wide Transcriptional Profiling Identifies BMP Signaling as Essential Regulator of Zebrafish Cardiomyocyte Regeneration. *Dev Cell*. 2016;36(1):36-49. doi:10.1016/j.devcel.2015.12.010

# Index

- \* **datasets**
  - zh.data, 20
- corHeatmap, 2
- createTomo, 3, 19, 20
- createTomo, matrix-method (createTomo), 3
- createTomo, missing-method (createTomo), 3
- createTomo, SummarizedExperiment-method (createTomo), 3
- dist, 12
- embedPlot, 5
- expHeatmap, 6
- findPeak, 7
- findPeakGene, 7
- geneCorHeatmap, 9
- geneEmbedPlot, 10
- hclust, 12
- hierarchClust, 11
- kmeans, 13
- kmeansClust, 12
- linePlot, 13
- normalizeTomo, 5, 14
- prcomp, 15
- Rtsne, 16
- runPCA, 15
- runTSNE, 16
- runUMAP, 17
- scaleTomo, 5, 18
- tomoMatrix, 4, 18
- tomoSummarizedExperiment, 4, 19
- umap, 17
- zh.data, 20