

# Package ‘tomoseqr’

May 30, 2024

**Title** R Package for Analyzing Tomo-seq Data

**Version** 1.8.0

**Description** `tomoseqr` is an R package for analyzing Tomo-seq data.

Tomo-seq is a genome-wide RNA tomography method that combines combining high-throughput RNA sequencing with cryosectioning for spatially resolved transcriptomics. `tomoseqr` reconstructs 3D expression patterns from tomo-seq data and visualizes the reconstructed 3D expression patterns.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** FALSE

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.2

**VignetteBuilder** knitr

**Depends** R (>= 4.2)

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

**Imports** grDevices, graphics, animation, tibble, dplyr, stringr, purrr, methods, shiny, BiocFileCache, readr, tools, plotly, ggplot2

**Suggests** rmarkdown, knitr, BiocStyle, testthat (>= 3.0.0)

**Config/testthat/edition** 3

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

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 8ed70d8

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-29

**Author** Ryosuke Matsuzawa [aut, cre] (<<https://orcid.org/0000-0003-2999-4734>>)

**Maintainer** Ryosuke Matsuzawa <[kinakomochi.work@gmail.com](mailto:kinakomochi.work@gmail.com)>

## Contents

animate2d . . . . .	2
doadJunker2014 . . . . .	3
downloadJunker2014 . . . . .	4
estimate3dExpressions . . . . .	4
extractGeneList . . . . .	5
findAxialGenes . . . . .	6
getReconstructedResult . . . . .	6
imageViewr . . . . .	7
mask . . . . .	7
masker . . . . .	8
plot1dAllExpression . . . . .	8
plot1dExpression . . . . .	9
plotLossFunction . . . . .	9
testx . . . . .	10
testy . . . . .	10
testz . . . . .	10
toDataFrame . . . . .	11
tomoObj . . . . .	11
<b>Index</b>	<b>12</b>

---

animate2d	<i>Animate 2D expressions along one axis and generate GIF file.</i>
-----------	---

---

## Description

Animate 2D expressions along one axis and generate GIF file.

## Usage

```
animate2d(
  tomoObj,
  geneID,
  along = "x",
  main = geneID,
  xlab = "x",
  ylab = "y",
  file = str_c(geneID, "_", along, ".gif"),
  zlim = NA,
  interval = 0.1,
  aspectX = 1,
  aspectY = 1
)
```

**Arguments**

tomoObj	tomoSeq object
geneID	single gene ID (string)
along	Parameter specifying along which axis the cross section should be plotted.
main	A string used for the title of the plot. Default is geneID.
xlab	Label of x axis. Default is xaxis.
ylab	Label of y axis. Default is yaxis.
file	Path of GIF file.
zlim	Limit of value of heatmap. If target="mask", it is ignored.
interval	interval of GIF animation.
aspectX	Width of figure. If you don't specify the value of this parameter, It is calculated based on the number of sections Corresponding to the horizontal axis
aspectY	Height of figure. If you don't specify the value of this parameter, It is calculated based on the number of sections Corresponding to the vertical axis

**Value**

It generate GIF file.

**Examples**

```
if(interactive()) {
  data(tomoObj)
  animate2d(tomoObj, "gene2", target = "expression", file = "example.gif")
}
```

---

doadJunker2014	<i>Load data of Junker2014 from cache.</i>
----------------	--

---

**Description**

Load data of Junker2014 from cache.

**Usage**

```
doadJunker2014(tomoseqrCache)
```

**Arguments**

tomoseqrCache Cache of tomoseqr. You can get it using downloadJunker2014.

**Value**

List of tomo-seq data in cache.

---

downloadJunker2014     *Download part of the Tomo-seq data published by Junker et al.*

---

**Description**

Download part of the Tomo-seq data published by Junker et al.

**Usage**

```
downloadJunker2014(verbose = FALSE)
```

**Arguments**

verbose                If you want to force downloads with or without cache, set this to TRUE.

**Value**

BiocFileCache object.

---

estimate3dExpressions     *Estimate 3d expressions*

---

**Description**

Estimate 3d expressions

**Usage**

```
estimate3dExpressions(x, y, z, mask, query, numIter = 100, normalize = TRUE)
```

**Arguments**

x	A data.frame object containing a simulated Tomo-seq data for x-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.
y	A data.frame object containing a simulated Tomo-seq data for y-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.
z	A data.frame object containing a simulated Tomo-seq data for z-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.
mask	A 3D array that represents if each boxel is included to sample. You can make a mask using masker.
query	Vector of gene IDs
numIter	How many times iterate
normalize	Whether to normalize so that total expression per sample volume is equal between sections.

**Value**

tomoSeq object

**Examples**

```
data("testx", "testy", "testz", "mask")
estimate3dExpressions(
  testx,
  testy,
  testz,
  mask = mask,
  query = c("gene1"),
  normalize = TRUE
)
```

---

extractGeneList	<i>Extract geneIDs to which hoge can be applied.</i>
-----------------	--

---

**Description**

Extract geneIDs to which hoge can be applied.

**Usage**

```
extractGeneList(x, y, z)
```

**Arguments**

- |   |  |
|---|--|
| x | A data.frame object containing a Tomo-seq data for x-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections. |
| y | A data.frame object containing a Tomo-seq data for y-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections. |
| z | A data.frame object containing a Tomo-seq data for z-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections. |

**Value**

A vector that contains genes which can be used for estimate3dExpressions.

**Examples**

```
data("testx", "testy", "testz")
extractGeneList(testx, testy, testz)
```

---

<code>findAxialGenes</code>	<i>Find peak genes on axial</i>
-----------------------------	---------------------------------

---

**Description**

Find peak genes on axial

**Usage**

```
findAxialGenes(tomoSeqData, genes = "all")
```

**Arguments**

<code>tomoSeqData</code>	tomo-seq data of any axis
<code>genes</code>	If run for all genes

**Value**

A data frame consisting of gene ID, max of expression levels of the gene, mean of expression levels calculated by excluding the maximum value and section number showing the maximum expression level (0 means that there is no such section).

**Examples**

```
data(testx)
findAxialGenes(testx)
```

---

<code>getReconstructedResult</code>	<i>Get reconstructed matrix</i>
-------------------------------------	---------------------------------

---

**Description**

Get reconstructed matrix

**Usage**

```
getReconstructedResult(tomoObj, geneID)
```

**Arguments**

<code>tomoObj</code>	tomoSeq object
<code>geneID</code>	single gene ID

**Value**

Reconstruction result as matrix

**Examples**

```
data(tomoObj)
getReconstructedResult(tomoObj, "gene2")
```

---

imageView	<i>Output the reconstructed expression pattern as an image.</i>
-----------	---

---

**Description**

Output the reconstructed expression pattern as an image.

**Usage**

```
imageView(tomoObj)
```

**Arguments**

tomoObj      tomoSeq object

**Value**

NA

**Examples**

```
if (interactive()) {
  data(tomoObj)
  imageView(tomoObj, "gene2")
}
```

---

mask	<i>A matrix containing mask data.</i>
------	---------------------------------------

---

**Description**

It is a matrix representing the mask. Each element of the matrix corresponds to a voxel, and has the value 1 if the voxel is included in the mask, 0 if it is not.

masker                      *Make mask*

---

**Description**

Make mask

**Usage**

```
masker()
```

**Value**

NA

**Examples**

```
if (interactive()) {  
  masker()  
}
```

---

plot1dAllExpression      *Plot expressions of all genes along an axis*

---

**Description**

Plot expressions of all genes along an axis

**Usage**

```
plot1dAllExpression(tomoSeqData, ...)
```

**Arguments**

tomoSeqData      A data.frame object containing a Tomo-seq data for any sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.

...                Arguments which are related to plot parameters. Please refer to [plot](#).

**Value**

NA

**Examples**

```
data("testx")  
plot1dAllExpression(testx)
```



---

plot1dExpression	<i>Plot expression of single gene along an axis</i>
------------------	---

---

**Description**

Plot expression of single gene along an axis

**Usage**

```
plot1dExpression(tomoObj, geneID, axes)
```

**Arguments**

tomoObj	tomoSeq object
geneID	single gene ID (string)
axes	axis ("x", "y" or "z")

**Value**

NA

**Examples**

```
data(tomoObj)
plot1dExpression(tomoObj, "gene2", "x")
```

---

plotLossFunction	<i>Plot the trend of the value of the loss function.</i>
------------------	--

---

**Description**

Plot the trend of the value of the loss function.

**Usage**

```
plotLossFunction(tomoObj, geneID)
```

**Arguments**

tomoObj	tomoSeq object
geneID	single gene ID (string)

**Value**

NA

**Examples**

```
data(tomoObj)
plotLossFunction(tomoObj, "gene2")
```

---

testx	<i>A data.frame object containing a simulated Tomo-seq data for x-axis sections.</i>
-------	--

---

**Description**

A data.frame object containing a simulated Tomo-seq data for x-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.

---

testy	<i>A data.frame object containing a simulated Tomo-seq data for y-axis sections.</i>
-------	--

---

**Description**

A data.frame object containing a simulated Tomo-seq data for y-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.

---

testz	<i>A data.frame object containing a simulated Tomo-seq data for z-axis sections.</i>
-------	--

---

**Description**

A data.frame object containing a simulated Tomo-seq data for z-axis sections. The rows represent genes. The first column contains gene IDs and the second and subsequent columns contain gene expression levels in sections.

---

toDataFrame	<i>Convert reconstructed matrix to data.frame.</i>
-------------	--

---

**Description**

Convert reconstructed matrix to data.frame.

**Usage**

```
toDataFrame(tomoObj, geneID)
```

**Arguments**

tomoObj	tomoSeq object
geneID	single gene ID

**Value**

Reconstruction result converted to dataframe.

**Examples**

```
data(tomoObj)
toDataFrame(tomoObj, "gene2")
```

---

tomoObj	<i>A tomoSeq object.</i>
---------	--------------------------

---

**Description**

It is a tomoSeq object generated from [testx](#), [testy](#), [testz](#) and [mask](#).

# Index

[animate2d](#), [2](#)

[doadJunker2014](#), [3](#)

[downloadJunker2014](#), [4](#)

[estimate3dExpressions](#), [4](#)

[extractGeneList](#), [5](#)

[findAxialGenes](#), [6](#)

[getReconstructedResult](#), [6](#)

[imageView](#), [7](#)

[mask](#), [7](#), [11](#)

[masker](#), [8](#)

[plot](#), [8](#)

[plot1dAllExpression](#), [8](#)

[plot1dExpression](#), [9](#)

[plotLossFunction](#), [9](#)

[testx](#), [10](#), [11](#)

[testy](#), [10](#), [11](#)

[testz](#), [10](#), [11](#)

[toDataFrame](#), [11](#)

[tomoObj](#), [11](#)