

# Package: ssviz (via r-universe)

May 30, 2026

**Type** Package

**Title** A small RNA-seq visualizer and analysis toolkit

**Version** 1.46.0

**Date** 2021-11-21

**Author** Diana Low

**Maintainer** Diana Low <lowdiana@gmail.com>

**Description** Small RNA sequencing viewer

**License** GPL-2

**Depends** R (>= 2.15.1), methods, Rsamtools, Biostrings, reshape, ggplot2, RColorBrewer, stats

**biocViews** ImmunoOncology, Sequencing, RNASeq, Visualization, MultipleComparison, Genetics

**Collate** AllClasses.R AllGenerics.R helper.R

**VignetteBuilder** knitr

**Suggests** knitr

**RoxygenNote** 6.0.1

**Config/pak/sysreqs** libbz2-dev liblzma-dev xz-utils zlib1g-dev

**Repository** <https://bioc-release.r-universe.dev>

**Date/Publication** 2026-04-28 12:39:18 UTC

**RemoteUrl** <https://github.com/bioc/ssviz>

**RemoteRef** RELEASE\_3\_23

**RemoteSha** e420a0fe4911d6d3198be7c32ef0e5adaeadeb963

## Contents

ssviz-package . . . . .	2
counts . . . . .	3
ctrlbam . . . . .	3
getCountMatrix . . . . .	3

getCountMatrix-methods . . . . .	4
logicalORmissing-class . . . . .	5
ntfreq . . . . .	5
ntfreq-methods . . . . .	6
pctrlbam . . . . .	6
pingpong . . . . .	7
pingpong-methods . . . . .	8
plotDistro . . . . .	8
plotDistro-methods . . . . .	9
plotFreq . . . . .	9
plotFreq-methods . . . . .	10
plotPP . . . . .	10
plotPP-methods . . . . .	11
plotRegion . . . . .	11
plotRegion-methods . . . . .	12
ptreatbam . . . . .	12
readBam . . . . .	12
readBam-methods . . . . .	13
treatbam . . . . .	14

**Index** **15**

---

ssviz-package	<i>ssviz</i>
---------------	--------------

---

**Description**

A package for short RNA seq visualization and quantification.

**Details**

Package: *ssviz*  
 Type: Package  
 Version: 0.99  
 Date: 2014-05-08  
 License: GPL-2

**Author(s)**

Diana H.P. Low Maintainer: Diana Low <dlow@imcb.a-star.edu.sg>

---

counts	<i>counts data</i>
--------	--------------------

---

**Description**

counts is an example total read count for bam reads

**Usage**

```
data(ssviz)
```

**Source**

internal

---

ctrlbam	<i>ctrlbam data</i>
---------	---------------------

---

**Description**

ctrlbam is an example control dataset from bam file read in with [readBam](#)

**Usage**

```
data(ssviz)
```

**Source**

internal

---

getCountMatrix	<i>getCountMatrix</i>
----------------	-----------------------

---

**Description**

returns the bam data.frame with an additional column counts. Only relevant if the fasta file used for mapping input was previously collapsed via fastx\_toolkit to return a fasta read name in the format of readnumber-totalcounts

**Usage**

```
getCountMatrix(bam_file,pseudo=FALSE)
```

**Arguments**

<code>bam_file</code>	An object of class <code>DataFrame</code> (from <code>IRanges</code> ). Can be generated from <code>readBam</code> .
<code>pseudo</code>	Logical. If <code>TRUE</code> , assume the reads in the bam file does not have a count record and sets all counts to 1.

**Value**

An object of class `data.frame` having the values from the original bam file with an additional 'count' column.

**Author(s)**

Diana H.P. Low

**See Also**

[readBam](#)

**Examples**

```
data(ssviz)
getCountMatrix(ctrlbam)
```

---

`getCountMatrix-methods`

*getCountMatrix*

---

**Description**

returns the bam `data.frame` with an additional column counts. Only relevant if the fasta file used for mapping input was previously collapsed via `fastx_toolkit` to return a fasta read name in the format of `readnumber-totalcounts`

**Methods**

`signature(object="DataFrame")` Returns and object of class `data.frame` having the values from the original bam file with an additional 'count' column.

---

```
logicalORmissing-class  
      Class "logicalORmissing"
```

---

**Description**

Class union of logical and missing object.

**Author(s)**

Diana H.P. Low

**Examples**

```
showClass("logicalORmissing")
```

---

```
ntfreq          ntfreq
```

---

**Description**

Calculates nucleotide frequency of reads in bam file

**Usage**

```
ntfreq(bam_file, ntlength, toRNA = TRUE, count_type = "total")
```

**Arguments**

bam_file	An object of class data.frame or DataFrame
ntlength	An integer specifying the length of the sequence to quantify
toRNA	A logical value on whether to translate the DNA sequence to RNA
count_type	A character string on how to count the nucleotides. Can be either "total" or "unique". If total is selected, the function will look for the countcolumn and multiply the reads by its number of occurrence when calculating the frequency.

**Value**

Returns a data.frame of the frequency of nucleotides (either A/C/G/T or A/C/G/U) at each position up to the specified ntlength

**Author(s)**

Diana H.P. Low

**Examples**

```
data(ssviz)
freq<-ntfreq(pctrlbam,ntlength=10)
```

---

ntfreq-methods	<i>ntfreq</i>
----------------	---------------

---

**Description**

Calculates nucleotide frequency of reads in bam file

**Methods**

`ntfreq(bam_file, ntlength, toRNA = TRUE, count_type = "total")` Returns a data frame of nucleotide frequencies along length of sequence provided.

**Author(s)**

Diana H.P. Low

---

pctrlbam	<i>pctrlbam data</i>
----------	----------------------

---

**Description**

pctrlbam is an example control dataset from bam file read in with [readBam](#)

**Usage**

```
data(ssviz)
```

**Source**

internal

---

pingpong

*pingpong*

---

## **Description**

piRNA ping-pong analysis of complementary sequences

## **Usage**

```
pingpong(bam_file)
```

## **Arguments**

`bam_file` An object of class `data.frame` or `DataFrame`

## **Details**

The ping-pong mechanism is a proposed method for the amplification of primary piRNAs, which leads to the production of new primary piRNAs from their precursor transcripts, which eventually amplifies the pool of both primary and secondary piRNAs. This positive feedback loop is a secondary biogenesis mechanism that requires complementary transcripts to a pre-existing pool of piRNAs.

## **Value**

This function returns a `data.frame` object with frequency of overlapping complementary piRNAs.

## **Author(s)**

Diana H.P. Low

## **References**

Brennecke J. et al. Cell 128, 1089-1103, March 23, 2007

## **Examples**

```
data(ssviz)
pp<-pingpong(pctrlbam)
```

---

pingpong-methods      *pingpong*

---

**Description**

piRNA ping-pong analysis of complementary sequences

**Methods**

`pingpong(bam_file)` Returns a data.frame object with frequency of overlapping complementary piRNAs.

**Author(s)**

Diana H.P. Low

---

plotDistro      *plotDistro*

---

**Description**

Plots distribution of reads in the bam file based on length, direction (strand) or location (rname)

**Usage**

`plotDistro(bamlist, type = "qwidth", samplenames = NULL, unique = FALSE, ncounts = NULL, norm = FALSE, yname = NULL)`

**Arguments**

<code>bamlist</code>	An object of type list, giving a list of bam files. If you only have 1 file, use <code>list(bam_file)</code>
<code>type</code>	An object of type character. Can be <code>qwidth</code> , <code>rname</code> or <code>strand</code> . In theory, any column property existing in the bam file can be used, but these 3 would be most meaningful.
<code>samplenames</code>	Labels for the plot.
<code>unique</code>	Logical value to use unique reads (TRUE) or all reads (FALSE)
<code>ncounts</code>	Number of total counts in the bam file, used if <code>unique</code> is set to FALSE.
<code>norm</code>	Logical value to determine if plot will be normalised.
<code>yname</code>	y axis label.

**Author(s)**

Diana H.P. Low

**Examples**

```
data(ssviz)
plotDistro(list(ctrlbam))
```

---

plotDistro-methods      *plotDistro*

---

**Description**

Plots distribution of reads in the bam file based on length, direction (strand) or location (rname)

**Methods**

```
plotDistro(bamlist, type = "qwidth", samplenames = NULL, unique = FALSE, ncounts = 1e+06, norm = FALSE, yna)
```

Returns a distribution plot.

**Author(s)**

Diana H.P. Low

---

plotFreq                      *plotFreq*

---

**Description**

Plots nucleotide frequency generated by [ntfreq](#)

**Usage**

```
plotFreq(freqvector, percentage = TRUE)
```

**Arguments**

freqvector	data.frame object generated by <a href="#">ntfreq</a>
percentage	Logical value to represent y-axis as percentage or frequency.

**Author(s)**

Diana H.P. Low

**See Also**

[ntfreq](#)

**Examples**

```
data(ssviz)
freq<-ntfreq(pctrlbam,ntlength=10)
plotFreq(freq)
```

---

plotFreq-methods      *plotFreq*

---

**Description**

Plots nucleotide frequency generated by [ntfreq](#)

**Methods**

`plotFreq(freqvector, percentage = TRUE)` Returns a frequency bar plot.

**Author(s)**

Diana H.P. Low

---

plotPP      *plotPP*

---

**Description**

Plots the ping-pong frequency of piRNA amplification

**Usage**

```
plotPP(pout, samplenames = NULL)
```

**Arguments**

`pout`      An object of type `data.frame` generated by [pingpong](#)  
`samplenames`      An object of type character for sample labels.

**Author(s)**

Diana H.P. Low

**References**

Brennecke J. et al. Cell 128, 1089-1103, March 23, 2007

**See Also**

[pingpong](#)

**Examples**

```
data(ssviz)
pp<-pingpong(pctrlbam)
plotPP(list(pp))
```

---

plotPP-methods	<i>plotPP</i>
----------------	---------------

---

**Description**

Plots the ping-pong frequency of piRNA amplification

**Methods**

`plotPP(pout, samplenames = NULL)` Returns the pingpong amplification plot.

**Author(s)**

Diana H.P. Low

---

plotRegion	<i>plotRegion</i>
------------	-------------------

---

**Description**

Plots the read density given a chromosome region.

**Usage**

`plotRegion(bamlist, region, howsmooth = 2, ncounts = NULL, samplenames = NULL)`

**Arguments**

<code>bamlist</code>	An object of type list, giving a list of bam files. If you only have 1 file, use <code>list(bam_file)</code>
<code>region</code>	An object of type character defining the region to plot. Eg. <code>chr1:1000-2000</code>
<code>howsmooth</code>	Numeric value controlling smoothness of the plot.
<code>ncounts</code>	Total number of reads for plot normalization.
<code>samplenames</code>	Sample names

**Value**

Returns the x and y components of the region's reads and plots the density.

**Author(s)**

Diana H.P. Low

**Examples**

```
data(ssviz)
region<-'chr1:3015526-3080526'
plotRegion(list(ctrlbam), region=region)
```

---

plotRegion-methods     *plotRegion*

---

### Description

Plots the read density given a chromosome region.

### Methods

`plotRegion(bamlist, region, howsmooth = 2, ncounts = NULL, samplenames = NULL)` Returns the x and y components of the region's reads and plots the density.

### Author(s)

Diana H.P. Low

---

ptreatbam     *ptreatbam data*

---

### Description

ptreatbam is an example treatment dataset from bam file read in with [readBam](#)

### Usage

`data(ssviz)`

### Source

internal

---

readBam     *readBam*

---

### Description

Reads a bam file through RSamtools, and converts it into a data frame of class `DataFrame`

### Usage

`readBam(file_name, tags = character(0))`

**Arguments**

`file_name` Character string of bam file location  
`tags` Bam tags to import into the data frame. By default it only takes the standard values if none are given.

**Details**

This function formalizes what had been described in the RSamtools documentation and makes it easier to compute the downstream functions in this package.

**Value**

Returns the bam file contents in a readable dataframe format.

**Author(s)**

Diana H.P. Low

**References**

RSamtools package

**Examples**

```
bam.files <- dir(system.file("extdata", package = "ssviz"), full = TRUE, patt = "bam$")
ctrlbam <- readBam(bam.files[1])
```

---

readBam-methods      *readBam*

---

**Description**

Reads a bam file through RSamtools, and converts it into a data frame of class DataFrame

**Methods**

`readBam(bam_file, tags = character(0))` Returns the bam file contents in a readable dataframe format.

**Author(s)**

Diana H.P. Low

---

treatbam

*treatbam data*

---

**Description**

treatbam is an example treatment dataset from bam file read in with [readBam](#)

**Usage**

```
data(ssviz)
```

**Source**

internal

# Index

- \* **classes**
  - logicalORmissing-class, 5
- \* **datasets**
  - counts, 3
  - ctrlbam, 3
  - pctrlbam, 6
  - ptreatbam, 12
  - treatbam, 14
- \* **methods**
  - ntfreq-methods, 6
  - pingpong-methods, 8
  - plotDistro-methods, 9
  - plotFreq-methods, 10
  - plotPP-methods, 11
  - plotRegion-methods, 12
  - readBam-methods, 13
- \* **package**
  - ssviz-package, 2
- counts, 3
- ctrlbam, 3
- getCountMatrix, 3
- getCountMatrix, DataFrame-method
  - (getCountMatrix-methods), 4
- getCountMatrix-methods, 4
- logicalORmissing-class, 5
- ntfreq, 5, 9, 10
- ntfreq, DataFrame, numeric-method
  - (ntfreq-methods), 6
- ntfreq-methods, 6
- pctrlbam, 6
- pingpong, 7, 10
- pingpong, DataFrame-method
  - (pingpong-methods), 8
- pingpong-methods, 8
- plotDistro, 8
- plotDistro, list-method
  - (plotDistro-methods), 9
- plotDistro-methods, 9
- plotFreq, 9
- plotFreq, data.frame, logicalORmissing-method
  - (plotFreq-methods), 10
- plotFreq-methods, 10
- plotPP, 10
- plotPP, list-method (plotPP-methods), 11
- plotPP-methods, 11
- plotRegion, 11
- plotRegion, list, character-method
  - (plotRegion-methods), 12
- plotRegion-methods, 12
- ptreatbam, 12
- readBam, 3, 4, 6, 12, 12, 14
- readBam, character-method
  - (readBam-methods), 13
- readBam-methods, 13
- ssviz (ssviz-package), 2
- ssviz-package, 2
- treatbam, 14