

# Package: BioVizSeq (via r-universe)

February 20, 2025

**Type** Package

**Title** Visualizing the Elements Within Bio-Sequences

**Version** 1.0.1

**Maintainer** Shiqi Zhao <zhaosq89@163.com>

**Description** Visualizing the types and distribution of elements within bio-sequences. At the same time, We have developed a geom layer, geom\_rrect(), that can generate rounded rectangles. No external references are used in the development of this package.

**Encoding** UTF-8

**License** Artistic-2.0

**Language** en-US

**RoxygenNote** 7.2.3

**Repository** CRAN

**Imports** dplyr, ggplot2, httr, RColorBrewer, seqinr, stringr, tidyr, magrittr, ggtree, treeio, shiny

**NeedsCompilation** no

**Author** Shiqi Zhao [aut, cre, cph]  
(<<https://orcid.org/0000-0002-2360-956X>>)

**Date/Publication** 2025-01-11 07:30:02 UTC

**Config/pak/sysreqs** make libicu-dev libssl-dev zlib1g-dev

## Contents

biovizseq . . . . .	2
cdd_plot . . . . .	3
cdd_to_loc . . . . .	4
combi_p . . . . .	4
fastaleng . . . . .	6
geom_rrect . . . . .	6
gff_plot . . . . .	8

gff_to_loc . . . . .	9
meme_plot . . . . .	10
meme_seq . . . . .	11
meme_to_loc . . . . .	12
motif_plot . . . . .	12
pfam_plot . . . . .	14
pfam_to_loc . . . . .	15
plantcare_classify . . . . .	16
plantcare_plot . . . . .	16
plantcare_statistic1 . . . . .	17
plantcare_statistic2 . . . . .	18
plantcare_to_loc . . . . .	19
smart_plot . . . . .	19
smart_to_loc . . . . .	20
upload_fa_to_plantcare . . . . .	21

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

---

biovizseq	<i>BioVizSeq shiny app start function.</i>
-----------	--

---

## Description

BioVizSeq shiny app start function.

## Usage

```
biovizseq()
```

## Value

Shinyapp: BioVizSeq shiny app.

## Author(s)

Shiqi Zhao

## Examples

```
# 1. Library BioVizSeq package
library(BioVizSeq)
```

---

cdd\_plot                      *cdd\_plot*

---

**Description**

Visualization of domain in CDD file

**Usage**

```
cdd_plot(  
  cdd_file,  
  fasta_file,  
  the_order = NULL,  
  domain_select = NULL,  
  shape = "RoundRect",  
  r = 0.3,  
  legend_size = 15,  
  domain_color = NULL  
)
```

**Arguments**

cdd_file	The path of cdd file.
fasta_file	The path of fasta file.
the_order	The path of order file. A List of Gene ID , One ID Per Line.
domain_select	The domain ID which you want to align with.
shape	RoundRect or Rect.
r	The radius of rounded corners.
legend_size	The size of legend.
domain_color	The color set of domain.

**Value**

p

**Author(s)**

Shiqi Zhao

**Examples**

```
hitdata_path <- system.file("extdata", "hitdata.txt", package = "BioVizSeq")  
fa_path <- system.file("extdata", "idpep.fa", package = "BioVizSeq")  
cdd_plot(hitdata_path, fa_path)
```

```
order_path <- system.file("extdata", "order.csv", package = "BioVizSeq")  
cdd_plot(hitdata_path, fa_path, the_order = order_path)
```

---

cdd_to_loc	<i>cdd_to_loc</i>
------------	-------------------

---

**Description**

Extract the location information of domain from cdd file

**Usage**

```
cdd_to_loc(cdd_file)
```

**Arguments**

cdd\_file      CDD file.

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
hitdata_path <- system.file("extdata", "hitdata.txt", package = "BioVizSeq")
cdd_file <- readLines(hitdata_path)
domain_loc <- cdd_to_loc(cdd_file)
```

---

combi_p	<i>combi_p</i>
---------	----------------

---

**Description**

Get ggplot2 files to facilitate free combination in patchwork

**Usage**

```
combi_p(  
  tree_path,  
  gff_path = NULL,  
  meme_path = NULL,  
  pfam_path = NULL,  
  cdd_path = NULL,  
  fa_path = NULL,  
  smart_path = FALSE,
```

```
    plantcare_path = NULL,  
    promoter_length = NULL,  
    renamefile = NULL,  
    shape = "RoundRect",  
    r = 0.3,  
    legend_size = 6  
  )
```

### Arguments

tree_path	The path of tree file (.newick).
gff_path	The path of .gff/gtf file.
meme_path	The path of .meme/mast file.
pfam_path	The path of pfam result file (.tsv).
cdd_path	The path of cdd result file (.txt).
fa_path	The path of protein file (.fa/fasta).
smart_path	Do SMART or not. (TRUE or FALSE)
plantcare_path	The path of plantcare file (.tab).
promoter_length	The length of promoter.
renamefile	Rename file. Two cils: new_name and old_name.
shape	RoundRect or Rect.
r	The radius of rounded corners.
legend_size	The size of legend.

### Value

list

### Author(s)

Shiqi Zhao

### Examples

```
tree_path <- system.file("extdata", "idpep.nwk", package = "BioVizSeq")  
plot_file <- combi_p(tree_path)
```

---

fastaleng	<i>fastaleng</i>
-----------	------------------

---

**Description**

Statistical sequence length

**Usage**

```
fastaleng(fasta_file)
```

**Arguments**

fasta\_file      The path of protein fasta file.

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
fasta_path <- system.file("extdata", "idpep.fa", package = "BioVizSeq")
fastaleng(fasta_path)
```

---

geom_rrect	<i>geom_rrect</i>
------------	-------------------

---

**Description**

Rounded rectangle

**Usage**

```
geom_rrect(  
  mapping = NULL,  
  data = NULL,  
  stat = "identity",  
  position = "identity",  
  r = 0.2,  
  ...,  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE  
)
```

**Arguments**

mapping	Set of aesthetic mappings created by <a href="#">aes</a> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame.
stat	Name of stat to modify data.
position	The position adjustment to use for overlapping points on this layer.
r	The radius of rounded corners.
...	additional parameter, e.g. <code>color</code> , <code>linewidth</code> , <code>alpha</code> .
na.rm	If "FALSE" (default), missing values are removed with a warning. If "TRUE", missing values are silently removed, logical.
show.legend	Whether to show legend, logical.
inherit.aes	Whether to inherit aesthetic mappings, logical, defaults to "TRUE".

**Details**

draws rounded rectangle by using the locations of the four corners (`xmin`, `xmax`, `ymin` and `ymax`) like `geom_rect()`.

**Value**

ggplot object

**Aesthetics**

`geom_rrect()` understands the following aesthetics (required aesthetics are in bold):

- `xmin`
- `xmax`
- `ymin`
- `ymax`
- **`alpha`**
- **`colour`**
- **`fill`**
- **`group`**
- **`linetype`**

Learn more about setting these aesthetics in `vignette("ggplot2-specs")`.

**Author(s)**

Shiqi Zhao

**Examples**

```

library(ggplot2)
df <- data.frame(
  xmin = c(1, 2, 3),
  xmax = c(2, 3, 4),
  ymin = c(1, 2, 3),
  ymax = c(2, 3, 4),
  category = c("A", "B", "C")
)

p <- ggplot(df) +
  geom_rrect(aes(xmin = xmin, xmax = xmax,
                ymin = ymin, ymax = ymax, fill = category),
            r = 0.4, linewidth = 1, colour = "black")

print(p)

```

---

*gff\_plot**gff\_plot*

---

**Description**

Visualization of element in gff or gtf file

**Usage**

```

gff_plot(
  gff_file,
  the_order = NULL,
  shape = "Rect",
  r = 0.3,
  legend_size = 15,
  element_color = NULL
)

```

**Arguments**

<code>gff_file</code>	The path of gff file.
<code>the_order</code>	The path of order of mRNA. It is also the mRNA you want to showcase. A List of Gene ID , One ID Per Line.
<code>shape</code>	RoundRect or Rect.
<code>r</code>	The radius of rounded corners.
<code>legend_size</code>	The size of legend.
<code>element_color</code>	The color set of element.



**Value**

p

**Author(s)**

Shiqi Zhao

**Examples**

```
gff_path <- system.file("extdata", "test.gff", package = "BioVizSeq")  
gff_plot(gff_path)
```

---

<i>gff_to_loc</i>	<i>gff_to_loc</i>
-------------------	-------------------

---

**Description**

Extract the location information of element from gff or gtf file

**Usage**

```
gff_to_loc(gff_data, mRNA_ID = NULL)
```

**Arguments**

*gff\_data*           gff file.  
*mRNA\_ID*           The mRNA you selected. If NULL, it means selecting all mRNAs.

**Value**

list

**Author(s)**

Shiqi Zhao

**Examples**

```
gff_path <- system.file("extdata", "test.gff", package = "BioVizSeq")  
gff_data <- read.table(gff_path, header = FALSE, sep = '\t')  
gff_loc <- gff_to_loc(gff_data)  
  
ID_path <- system.file("extdata", "ID_select.csv", package = "BioVizSeq")  
mRNA_ID <- readLines(ID_path)  
gff_loc <- gff_to_loc(gff_data, mRNA_ID=mRNA_ID)
```

---

`meme_plot`*meme\_plot*

---

**Description**

Visualization of motif in meme file or mast file

**Usage**

```
meme_plot(  
  meme_file,  
  the_order = NULL,  
  motif_select = NULL,  
  shape = "RoundRect",  
  show_motif_id = FALSE,  
  r = 0.3,  
  legend_size = 15,  
  motif_color = NULL  
)
```

**Arguments**

<code>meme_file</code>	The path of meme file or mast file.
<code>the_order</code>	The path of order file. A List of Gene ID , One ID Per Line.
<code>motif_select</code>	The motif ID which you want to align with.
<code>shape</code>	RoundRect or Rect.
<code>show_motif_id</code>	Display the name of the motif.
<code>r</code>	The radius of rounded corners.
<code>legend_size</code>	The size of legend.
<code>motif_color</code>	The color set of motif.

**Value**

p

**Author(s)**

Shiqi Zhao

**Examples**

```
meme_path <- system.file("extdata", "meme.xml", package = "BioVizSeq")  
meme_plot(meme_path)  
  
mast_path <- system.file("extdata", "mast.xml", package = "BioVizSeq")  
meme_plot(mast_path)
```

```
meme_plot(meme_path, motif_select="1", show_motif_id = TRUE)

order_path <- system.file("extdata", "order.csv", package = "BioVizSeq")
meme_plot(meme_path, the_order=order_path, motif_select="1")
```

---

meme\_seq

*meme\_seq*

---

## Description

Get motif sequence from meme file or mast file

## Usage

```
meme_seq(meme_file)
```

## Arguments

meme\_file      The path of meme file or mast file.

## Value

data.frame

## Author(s)

Shiqi Zhao

## Examples

```
meme_path <- system.file("extdata", "meme.xml", package = "BioVizSeq")
meme_file <- readLines(meme_path)
motifseq<- meme_seq(meme_file)

mast_path <- system.file("extdata", "mast.xml", package = "BioVizSeq")
mast_file <- readLines(mast_path)
motifseq<- meme_seq(mast_file)
```

---

meme_to_loc	<i>meme_to_loc</i>
-------------	--------------------

---

**Description**

Extract the location information of motif from mast or meme file

**Usage**

```
meme_to_loc(motif_file)
```

**Arguments**

motif\_file      The motif data of mast or meme file.

**Value**

list

**Author(s)**

Shiqi Zhao

**Examples**

```
meme_path <- system.file("extdata", "meme.xml", package = "BioVizSeq")
meme_file <- readLines(meme_path)
motif_loc <- meme_to_loc(meme_file)

mast_path <- system.file("extdata", "mast.xml", package = "BioVizSeq")
mast_file <- readLines(mast_path)
motif_loc <- meme_to_loc(mast_file)
```

---

motif_plot	<i>motif_plot</i>
------------	-------------------

---

**Description**

Draws multiple rounded rectangle.

**Usage**

```
motif_plot(
  motif_loc,
  gene_length,
  the_order = NULL,
  motif_select = NULL,
  shape = "RoundRect",
  show_motif_id = FALSE,
  r = 0.3,
  legend_size = 15,
  motif_color = NULL
)
```

**Arguments**

motif_loc	A data.frame contains for columns: ID, motif, start, end.
gene_length	A data.frame of the length of biosequences. Two columns: ID, length.
the_order	A List of Gene ID , One ID Per Line.
motif_select	The motif ID which you want to align with.
shape	RoundRect or Rect.
show_motif_id	Display the name of the motif.
r	The radius of rounded corners.
legend_size	The size of legend.
motif_color	The color set of motif.

**Details**

motif\_plot() draws multiple rounded rectangle to represent the above elements of biosequences, but not limited to biosequences

**Value**

p

**Author(s)**

Shiqi Zhao

**Examples**

```
df <- data.frame(
  ID = rep(c("geneA", "geneB", "geneC"), each = 3),
  motif = rep(c("1", "2", "3"), times = 3),
  start = c(1, 3, 6, 1, 6, 10, 10, 7, 17),
  end = c(3, 5, 11, 3, 8, 15, 12, 9, 22)
)
```

```
length_data <- data.frame(  
  ID = c("geneA", "geneB", "geneC"),  
  length = c(15, 27, 30)  
)  
  
order_data <- c("geneB", "geneA", "geneC")  
  
motif_plot(df, length_data)  
motif_plot(df, length_data, the_order = order_data)
```

---

pfam\_plot

*pfam\_plot*

---

## Description

Visualization of domain in pfam result file

## Usage

```
pfam_plot(  
  pfam_file,  
  the_order = NULL,  
  domain_select = NULL,  
  shape = "RoundRect",  
  r = 0.3,  
  legend_size = 15,  
  domain_color = NULL  
)
```

## Arguments

pfam_file	The path of meme file or mast file.
the_order	The path of order file. A List of Gene ID , One ID Per Line.
domain_select	The domain ID which you want to align with.
shape	RoundRect or Rect.
r	The radius of rounded corners.
legend_size	The size of legend.
domain_color	The color set of domain.

## Value

P

## Author(s)

Shiqi Zhao

**Examples**

```
pfam_path <- system.file("extdata", "iprscan.tsv", package = "BioVizSeq")
order_path <- system.file("extdata", "order.csv", package = "BioVizSeq")
pfam_plot(pfam_path)
pfam_plot(pfam_path, the_order=order_path)
```

---

pfam_to_loc	<i>pfam_to_loc</i>
-------------	--------------------

---

**Description**

Extract the location information of domain from pfam result

**Usage**

```
pfam_to_loc(pfam_data)
```

**Arguments**

pfam\_data      The result file (.tsv) of pfam (via InterPro).

**Value**

list

**Author(s)**

Shiqi Zhao

**Examples**

```
pfam_path <- system.file("extdata", "iprscan.tsv", package = "BioVizSeq")
pfam_file <- read.table(pfam_path, sep='\t', header = FALSE)
motif_loc <- pfam_to_loc(pfam_file)
```

---

plantcare\_classify     *plantcare\_classify*

---

**Description**

Classify the functions of cis element from Plantcare

**Usage**

```
plantcare_classify(plantcare_file)
```

**Arguments**

plantcare\_file The result file (.tab) of Plantcare.

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
plantcare_path <- system.file("extdata", "plantCARE_output.tab", package = "BioVizSeq")
plantcare_file <- read.table(plantcare_path, header = FALSE, sep = '\t', quote="")
plantcare_data <- plantcare_classify(plantcare_file)
```

---

plantcare\_plot     *plantcare\_plot*

---

**Description**

Visualization of cis-element in plantcare result file

**Usage**

```
plantcare_plot(
  plantcare_file,
  promoter_length = 2000,
  the_order = NULL,
  shape = "Rect",
  r = 6,
  legend_size = 15,
  element_color = NULL
)
```



**Arguments**

plantcare\_file The path of plantcare result file (.tab).  
promoter\_length The promoter length.  
the\_order The path of order file. A List of Gene ID , One ID Per Line.  
shape RoundRect or Rect.  
r The radius of rounded corners.  
legend\_size The size of legend.  
element\_color The color set of cis-element.

**Value**

p

**Author(s)**

Shiqi Zhao

**Examples**

```
plantcare_path <- system.file("extdata", "plantCARE_output.tab", package = "BioVizSeq")  
plantcare_plot(plantcare_path, promoter_length = 2000)
```

---

plantcare\_statistic1 *plantcare\_statistic1*

---

**Description**

Count the number of cis element from Plantcare for heatmap

**Usage**

```
plantcare_statistic1(plantcare_data)
```

**Arguments**

plantcare\_data The result of plantcare\_classify().

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
plantcare_path <- system.file("extdata", "plantCARE_output.tab", package = "BioVizSeq")
plantcare_file <- read.table(plantcare_path, header = FALSE, sep = '\t', quote="")
plantcare_data <- plantcare_classify(plantcare_file)
statistic_data1 <- plantcare_statistic1(plantcare_data)
```

---

plantcare\_statistic2 *plantcare\_statistic2*

---

**Description**

Count the number of cis element from Plantcare for Bar chart

**Usage**

```
plantcare_statistic2(plantcare_data)
```

**Arguments**

plantcare\_data The result of plantcare\_classify().

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
plantcare_path <- system.file("extdata", "plantCARE_output.tab", package = "BioVizSeq")
plantcare_file <- read.table(plantcare_path, header = FALSE, sep = '\t', quote="")
plantcare_data <- plantcare_classify(plantcare_file)
statistic_data2 <- plantcare_statistic2(plantcare_data)
```

---

plantcare_to_loc	<i>plantcare_to_loc</i>
------------------	-------------------------

---

**Description**

Extract the location information of cis-element from Plantcare

**Usage**

```
plantcare_to_loc(plantcare_data)
```

**Arguments**

plantcare\_data The result of plantcare\_classify().

**Value**

data.frame

**Author(s)**

Shiqi Zhao

**Examples**

```
plantcare_path <- system.file("extdata", "plantCARE_output.tab", package = "BioVizSeq")
plantcare_file <- read.table(plantcare_path, header = FALSE, sep = '\t', quote="")
plantcare_data <- plantcare_classify(plantcare_file)
plantcare_loc <- plantcare_to_loc(plantcare_data)
```

---

smart_plot	<i>smart_plot</i>
------------	-------------------

---

**Description**

Visualization of domain in SMART result file

**Usage**

```
smart_plot(
  fasta_file,
  the_order = NULL,
  domain_select = NULL,
  shape = "RoundRect",
  r = 0.3,
  legend_size = 15,
  domain_color = NULL
)
```

**Arguments**

fasta_file	The path of protein fasta file.
the_order	The path of order file. A List of Gene ID , One ID Per Line.
domain_select	The domain ID which you want to align with.
shape	RoundRect or Rect.
r	The radius of rounded corners.
legend_size	The size of legend.
domain_color	The color set of domain.

**Value**

p

**Author(s)**

Shiqi Zhao

---

smart\_to\_loc

*smart\_to\_loc*

---

**Description**

Extract the location information of domain from SMART result

**Usage**

```
smart_to_loc(input_file, do_pfam = TRUE)
```

**Arguments**

input_file	The path of protein fasta file.
do_pfam	Include the pfam domain or not.

**Value**

list

**Author(s)**

Shiqi Zhao

---

`upload_fa_to_plantcare`  
*upload\_fa\_to\_plantcare*

---

### **Description**

Upload the promoter file to Plantcare database

### **Usage**

`upload_fa_to_plantcare(fasta_file, email)`

### **Arguments**

<code>fasta_file</code>	The path of promoter file.
<code>email</code>	e-mail address.

### **Details**

Due to the file size limitation of plantcare on fasta, `upload_fa_to_plantcare()` first splits fasta file. Then uploads the splitted fasta files to the plantcare database, and automatically returns the results to the email provided by the user.

`upload_fa_to_plantcare("the path/test.fasta", "your e-mail address")`

### **Value**

`plantcare_result`

### **Author(s)**

Shiqi Zhao

# Index

aes, 7  
alpha, 7  
  
biovizseq, 2  
  
cdd\_plot, 3  
cdd\_to\_loc, 4  
colour, 7  
combi\_p, 4  
  
fastaleng, 6  
fill, 7  
  
geom\_rrect, 6  
gff\_plot, 8  
gff\_to\_loc, 9  
group, 7  
  
linetype, 7  
  
meme\_plot, 10  
meme\_seq, 11  
meme\_to\_loc, 12  
motif\_plot, 12  
  
pfam\_plot, 14  
pfam\_to\_loc, 15  
plantcare\_classify, 16  
plantcare\_plot, 16  
plantcare\_statistic1, 17  
plantcare\_statistic2, 18  
plantcare\_to\_loc, 19  
  
smart\_plot, 19  
smart\_to\_loc, 20  
  
upload\_fa\_to\_plantcare, 21  
  
xmax, 7  
xmin, 7  
  
ymax, 7  
ymin, 7