

# Package ‘tidygate’

May 8, 2026

**Type** Package

**Title** Interactively Gate Points

**Version** 1.0.19

**Maintainer** William Hutchison <hutchison.w@wehi.edu.au>

**Description** Interactively gate points on a scatter plot. Interactively drawn gates are recorded and can be applied programmatically to reproduce results exactly. Programmatic gating is based on the package gatepoints by Wajid Jawaid.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1.0)

**Imports** utils, graphics, lifecycle, scales, magrittr, tibble, dplyr, purrr, rlang, tidyr, viridis, grDevices, RColorBrewer, stringr, shiny, plotly, ggplot2, xfun

**RdMacros** lifecycle

**Suggests** testthat, markdown, knitr, readr

**VignetteBuilder** knitr

**Biarch** true

**biocViews** AssayDomain, Infrastructure

**URL** <https://github.com/stemangiola/tidygate>

**BugReports** <https://github.com/stemangiola/tidygate/issues>

**NeedsCompilation** no

**Author** Stefano Mangiola [aut],  
Wajid Jawaid [ctb],  
William Hutchison [cre, aut] (ORCID:  
<<https://orcid.org/0009-0001-6242-4269>>)

**Repository** CRAN

**Date/Publication** 2025-12-10 08:10:02 UTC

## Contents

demo_gate_data . . . . .	2
fhs . . . . .	2
gate . . . . .	3
gate_chr . . . . .	5
gate_chr.numeric . . . . .	7
gate_int.numeric . . . . .	8
gate_interactive . . . . .	9
gate_programmatic . . . . .	9
server . . . . .	10
ui . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

demo_gate_data	<i>Demo gate data</i>
----------------	-----------------------

---

### Description

Demo gate data

### Usage

demo\_gate\_data

### Format

An object of class tbl\_df (inherits from tbl, data.frame) with 26 rows and 3 columns.

---

fhs	<i>Freehand select</i>
-----	------------------------

---

### Description

Freehand select

### Usage

fhs(data, mark = TRUE, names = TRUE, ...)

### Arguments

data	Data frame or matrix of co-ordinates. (x,y) co-ordinates for each point will be on rows. Rownames of selected points will be returned.
mark	Default TRUE. Predicate marking of selected points.
names	Default TRUE. If TRUE will return rownames of data frame with points within polygon. If FALSE will return logical vector.
...	Additional parameters passed to <a href="#">points</a> .

## Details

Freehand select function. First generate a 2D plot using R's plot function, then select gate region by left clicking. Close polygon by right clicking. The function will return the rownames of the enclosed points by the rownames of the co-ordinates given in data.

## Value

Returns character vector of rownames of the selected points from data if names parameter is TRUE. If names is FALSE then a logical vector indicating whether points are in the polygon is returned.

## Author(s)

Wajid Jawaid

## Examples

```
if(interactive()) {  
  x <- cbind(1:10, 1:10)  
  rownames(x) <- 1:10  
  plot(x, pch = 16, col = "red")  
  fhs(x)  
}
```

---

gate

*Gate points*

---

## Description

Gate points based on their X and Y coordinates. By default, this function launches an interactive scatter plot. Colour, shape, size and alpha can be defined as constant values, or can be controlled by the values of a specified column.

If previously drawn gates are supplied to the 'programmatic\_gates' argument, points will be gated programmatically. This feature allows the reproduction of previously drawn interactive gates. Programmatic gating is based on the package gatepoints by Wajid Jawaid.

## Usage

```
gate(  
  x,  
  y,  
  colour = NULL,  
  shape = NULL,  
  alpha = 1,  
  size = 2,  
  rasterise_points = FALSE,  
  programmatic_gates = NULL  
)
```

## Arguments

x	A vector representing the X dimension.
y	A vector representing the Y dimension.
colour	A single colour code string compatible with ggplot2. Or, a vector representing the point colour.
shape	A single ggplot2 shape numeric ranging from 0 to 127. Or, a vector representing the point shape, coercible to a factor of 6 or less levels.
alpha	A single ggplot2 alpha numeric ranging from 0 to 1. Or, a vector representing the point alpha, either a numeric or factor of 6 or less levels.
size	A single ggplot2 size numeric ranging from 0 to 20. Or, a vector representing the point size, either a numeric or factor of 6 or less levels.
rasterise_points	A logical. If TRUE, points are rasterised to an image before interactive gating is launched, improving performance for large datasets. Some interactive features are unavailable with rasterisation enabled.
programmatic_gates	A 'data.frame' of the gate brush data, as saved in 'tidygate_env\$gates'. The column 'x' records X coordinates, the column 'y' records Y coordinates and the column '.gate' records the gate number. When this argument is supplied, gates will be drawn programmatically.

## Value

A vector of strings, of the gates each X and Y coordinate pair is within. If gates are drawn interactively, they are temporarily saved to 'tidygate\_env\$gates'.

## Examples

```
library(dplyr)
data("demo_gate_data", package = "tidygate")

# Gate points interactively
if(interactive()) {
  mtcars |>
    mutate(gated = gate(x = mpg, y = wt, shape = am))
}

# Gate points programmatically
mtcars |>
  mutate(gated = gate(x = mpg, y = wt, programmatic_gates = demo_gate_data))
```

---

gate_chr	<i>Label points within a scatter plot drawing a gate</i>
----------	--

---

### Description

gate() takes as input a 'tbl' formatted as | <DIMENSION 1> | <DIMENSION 2> | <...> | and calculates the rotated dimensional space of the feature value.

### Usage

```
gate_chr(
  .dim1,
  .dim2,
  .color = NULL,
  .shape = NULL,
  .size = NULL,
  opacity = 1,
  how_many_gates = 1,
  .group_by = NULL,
  gate_list = NULL,
  ...
)
```

```
gate_int(
  .dim1,
  .dim2,
  .color = NULL,
  .shape = NULL,
  .size = NULL,
  opacity = 1,
  how_many_gates = 1,
  .group_by = NULL,
  gate_list = NULL,
  ...
)
```

### Arguments

.dim1	A column symbol. The x dimension
.dim2	A column symbol. The y dimension
.color	A column symbol. Colour of points
.shape	A column symbol. Shape of points
.size	A column symbol. Size of points
opacity	A number between 0 and 1. The opacity level of the data points
how_many_gates	An integer. The number of gates to label

<code>.group_by</code>	A column symbol. The column that is used to calculate distance (i.e., normally genes)
<code>gate_list</code>	A list of gates. It is returned by gate function as attribute <code>"gate"</code> . If you want to create this list yourself, each element of the list is a data frame with x and y columns. Each row is a coordinate. The order matter.
<code>...</code>	Further parameters passed to the function <code>gatepoints::fhs</code>

## Details

### [Maturing]

This function allow the user to label data points in inside one or more 2D gates. This package is based on on the package `gatepoints`.

## Value

An character vector, with "0" for elements outside gates and "1..N" for the elements inside the N gates.

An integer vector, with 0 for elements outside gates and 1..N for the elements inside the N gates.

## Examples

```
# Standard use - interactive

if(interactive()){

  tidygate::tidygate_data %>%
    distinct(`ct 1`, `ct 2`, Dim1, Dim2) %>%
    mutate(gate = gate_chr( Dim1, Dim2))

}

library(magrittr)
library(dplyr)

# Standard use - programmatic
res_distinct =
  tidygate::tidygate_data %>%
  distinct(`ct 1`, `ct 2`, Dim1, Dim2) %>%
  mutate(gate = gate_chr( Dim1, Dim2, gate_list = tidygate::gate_list))

# Grouping - programmatic
res =
  tidygate::tidygate_data %>%
  mutate(gate = gate_chr(
    Dim1, Dim2,
    .group_by = c(`ct 1`, `ct 2`),
    gate_list = tidygate::gate_list
  ))
```

---

gate_chr.numeric	<i>gate_chr</i>
------------------	-----------------

---

## Description

gate\_chr

## Usage

```
## S3 method for class 'numeric'
gate_chr(
  .dim1,
  .dim2,
  .color = NULL,
  .shape = NULL,
  .size = NULL,
  opacity = 1,
  how_many_gates = 1,
  .group_by = NULL,
  gate_list = NULL,
  ...
)
```

## Arguments

<code>.dim1</code>	A column symbol. The x dimension
<code>.dim2</code>	A column symbol. The y dimension
<code>.color</code>	A column symbol. Colour of points
<code>.shape</code>	A column symbol. Shape of points
<code>.size</code>	A column symbol. Size of points
<code>opacity</code>	A number between 0 and 1. The opacity level of the data points
<code>how_many_gates</code>	An integer. The number of gates to label
<code>.group_by</code>	A column symbol. The column that is used to calculate distance (i.e., normally genes)
<code>gate_list</code>	A list of gates. It is returned by gate function as attribute <code>"gate"</code> . If you want to create this list yourself, each element of the list is a data frame with x and y columns. Each row is a coordinate. The order matter.
<code>...</code>	Further parameters passed to the function <code>gatepoints::fhs</code>

## Value

An character vector, with "0" for elements outside gates and "1..N" for the elements inside the N gates.

---

gate\_int.numeric      *gate\_int*

---

## Description

gate\_int

## Usage

```
## S3 method for class 'numeric'
gate_int(
  .dim1,
  .dim2,
  .color = NULL,
  .shape = NULL,
  .size = NULL,
  opacity = 1,
  how_many_gates = 1,
  .group_by = NULL,
  gate_list = NULL,
  ...
)
```

## Arguments

.dim1	A column symbol. The x dimension
.dim2	A column symbol. The y dimension
.color	A column symbol. Colour of points
.shape	A column symbol. Shape of points
.size	A column symbol. Size of points
opacity	A number between 0 and 1. The opacity level of the data points
how_many_gates	An integer. The number of gates to label
.group_by	A column symbol. The column that is used to calculate distance (i.e., normally genes)
gate_list	A list of gates. It is returned by gate function as attribute <code>"gate"</code> . If you want to create this list yourself, each element of the list is a data frame with x and y columns. Each row is a coordinate. The order matter.
...	Further parameters passed to the function <code>gatepoints::fhs</code>

## Value

An integer vector, with 0 for elements outside gates and 1..N for the elements inside the N gates.

---

gate_interactive	<i>Interactively gate data with a simple scatter plot</i>
------------------	---

---

### Description

Create an interactive scatter plot based on user-defined X and Y coordinates. Colour, shape, size and alpha can be defined as constant values, or can be controlled by values in a specified column.

### Usage

```
gate_interactive(x, y, colour, shape, alpha, size, rasterise_points)
```

### Arguments

x	A vector representing the X dimension.
y	A vector representing the Y dimension.
colour	A single colour code string compatible with ggplot2. Or, a vector representing the point colour.
shape	A single ggplot2 shape numeric ranging from 0 to 127. Or, a vector representing the point shape, coercible to a factor of 6 or less levels.
alpha	A single ggplot2 alpha numeric ranging from 0 to 1. Or, a vector representing the point alpha, either a numeric or factor of 6 or less levels.
size	A single ggplot2 size numeric ranging from 0 to 20. Or, a vector representing the point size, either a numeric or factor of 6 or less levels.
rasterise_points	A logical. If TRUE, points are rasterised to an image before interactive gating is launched, improving performance for large datasets. Some interactive features are unavailable with rasterisation enabled.

### Value

A vector of strings, of the gates each X and Y coordinate pair is within. If gates are drawn interactively, they are temporarily saved to 'tidygate\_env\$gates'

---

gate_programmatic	<i>Programmatically gate data with pre-recorded lasso selection coordinates</i>
-------------------	---

---

### Description

A helpful way to repeat previous interactive lasso selections to enable reproducibility. Programmatic gating is based on the package [gatepoints](<https://github.com/wjawaid/gatepoints>) by Wajid Jawaid.

**Usage**

```
gate_programmatic(x, y, programmatic_gates)
```

**Arguments**

x                    A vector representing the X dimension.

y                    A vector representing the Y dimension.

programmatic\_gates

A 'data.frame' of the gate brush data, as saved in 'tidygate\_env\$gates'. The column 'x' records X coordinates, the column 'y' records Y coordinates and the column '.gate' records the gate number.

**Value**

A vector of strings, of the gates each X and Y coordinate pair is within.

---

server

*Run Shiny App for interactive gating*

---

**Description**

Run Shiny App for interactive gating

**Usage**

```
server(input, output, session)
```

**Arguments**

input                Server input parameter

output               Server output parameter

session              Server session parameter

**Value**

NA

---

`ui`*Create Shiny App UI*

---

**Description**

Create Shiny App UI

**Usage**

`ui`

**Format**

An object of class `shiny.tag.list` (inherits from `list`) of length 4.

**Value**

Fluid UI container

# Index

\* **datasets**

ui, [11](#)

\* **data**

demo\_gate\_data, [2](#)

demo\_gate\_data, [2](#)

fhs, [2](#)

gate, [3](#)

gate\_chr, [5](#)

gate\_chr.numeric, [7](#)

gate\_int(gate\_chr), [5](#)

gate\_int.numeric, [8](#)

gate\_interactive, [9](#)

gate\_programmatic, [9](#)

points, [2](#)

server, [10](#)

ui, [11](#)