

Package: ggseg (via r-universe)

July 3, 2026

Title Plotting Tool for Brain Atlases

Version 2.2.1

Description Provides a 'ggplot2' geom and position for visualizing brain region data on cortical, subcortical, and white matter tract atlases. Brain atlas geometries are stored as polygon data, enabling seamless integration with the 'ggplot2' ecosystem including faceting, custom scales, and themes. Mowinckel & Vidal-Piñeiro (2020) <[doi:10.1177/2515245920928009](https://doi.org/10.1177/2515245920928009)>.

License MIT + file LICENSE

URL <https://ggsegverse.github.io/ggseg/>,
<https://github.com/ggsegverse/ggseg>

BugReports <https://github.com/ggsegverse/ggseg/issues>

Depends R (>= 3.3)

Imports cli, dplyr (>= 1.0.0), ggplot2 (>= 3.3), ggseg.formats (>= 0.0.3), grid, lifecycle, rlang, tidyr (>= 1.0.0), utils

Suggests covr, devtools, here, knitr, rmarkdown, sf, spelling, testthat (>= 3.0.0), vdiff, withr

VignetteBuilder knitr

Config/Needs/website ggsegverse/ggseg.docs

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation no

Author Athanasia Mo Mowinckel [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5756-0223>>), Didac Vidal-Piñeiro [aut] (ORCID: <<https://orcid.org/0000-0001-9997-9156>>), Ramiro Magno [ctb] (ORCID: <<https://orcid.org/0000-0001-5226-3441>>),

Center for Lifespan Changes in Brain and Cognition, University
of Oslo, Norway [cph]

Maintainer Athanasia Mo Mowinckel <a.m.mowinckel@psykologi.uio.no>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-03 02:50:06 UTC

RemoteUrl <https://github.com/cran/ggseg>

RemoteRef HEAD

RemoteSha 72245fd50101f22b00d27ab8c958cb5d3ae7c047

Contents

annotate_brain	2
brain_join	4
coord_brain	5
geom_brain	6
position_brain	7
reposition_brain	9
scale_brain	10
scale_brain_manual	11
scale_brain2	12
scale_continuous_brain	12
theme_brain	13

Index	16
--------------	-----------

annotate_brain	<i>Add view labels to brain atlas plots</i>
----------------	---

Description

Annotates each brain view with a text label positioned above the view's bounding box. For cortical atlases, labels show hemisphere and view (e.g., "left lateral"). For subcortical and tract atlases, labels show the view name directly (e.g., "axial_1", "sagittal").

Usage

```
annotate_brain(
  atlas,
  position = position_brain(),
  hemi = NULL,
  view = NULL,
  size = 3,
  colour = "grey30",
  family = "mono",
  padding = 0.05,
```

```

    nudge_y = 0,
    ...
  )

```

Arguments

atlas	A 'brain_atlas' object (e.g. 'dk()', 'aseg()').
position	The same layout you passed to [geom_brain()], from [position_brain()].
hemi	Character vector of hemispheres to include. If 'NULL' (default), all hemispheres are included.
view	Character vector of views to include. If 'NULL' (default), all views are included.
size	Text size in mm (default: '3').
colour	Text colour (default: "grey30").
family	Font family (default: "mono").
padding	Vertical gap between each label and its view, as a fraction of the plot's total height (default: '0.05'). Labels are also bottom-anchored ('vjust = 0') so they sit clear of the geometry.
nudge_y	Additional absolute vertical offset for labels (default: '0').
...	Additional arguments passed to [ggplot2::annotate()].

Details

Pass the same 'position' you gave [geom_brain()] and the labels line up with the views automatically.

Value

A ggplot2 annotation layer.

Examples

```

library(ggplot2)

pos <- position_brain(hemi ~ view)
ggplot() +
  geom_brain(atlas = dk(), position = pos, show.legend = FALSE) +
  annotate_brain(atlas = dk(), position = pos)

ggplot() +
  geom_brain(atlas = dk(), show.legend = FALSE) +
  annotate_brain(atlas = dk())

```

`brain_join`*Join user data with a brain atlas*

Description

Matches your data to a brain atlas by a shared column (usually ‘region’), keeping every atlas region whether or not you have a value for it. Grouped data (via `[dplyr::group_by()]`) gives one complete atlas per group. You rarely need this directly – `[geom_brain()]` joins your data for you; reach for ‘`brain_join()`’ when you want the joined sf object yourself.

Usage

```
brain_join(data, atlas, by = NULL)
```

Arguments

<code>data</code>	A data.frame with a column matching an atlas column (typically “region”). Can be grouped with <code>[dplyr::group_by()]</code> .
<code>atlas</code>	A ‘ggseg_atlas’ object or data.frame containing atlas data.
<code>by</code>	Character vector of column names to join by. If ‘NULL’ (default), columns are detected automatically.

Value

An ‘sf’ object if the atlas contains geometry, otherwise a tibble.

Examples

```
someData <- data.frame(  
  region = c(  
    "transverse temporal", "insula",  
    "precentral", "superior parietal"  
  ),  
  p = sample(seq(0, .5, .001), 4),  
  stringsAsFactors = FALSE  
)  
  
brain_join(someData, dk())  
brain_join(someData, dk(), "region")
```

coord_brain	<i>Keep brain shapes undistorted</i>
-------------	--------------------------------------

Description

Fixes the aspect ratio so brains aren't stretched by the shape of the plotting window. [geom_brain()] adds 'coord_brain()' for you, so you rarely need to call it yourself – reach for it only to adjust the 'ratio' or 'clip' behaviour. You can safely add your own coord or stack several 'geom_brain()' layers; it steps aside cleanly.

Usage

```
coord_brain(ratio = 1, clip = "off", ...)
```

Arguments

ratio	Aspect ratio, expressed as 'y / x'. Defaults to '1', which keeps brain polygons undistorted.
clip	Should drawing be clipped to the panel extent ("on") or allowed to overflow ("off")? Defaults to "off" so region outlines at the panel edge are not cut.
...	Additional arguments passed to [ggplot2::coord_fixed()].

Value

A ggplot2 coordinate system that registers as a default.

Examples

```
library(ggplot2)
## Not run:
poly <- ggseg.formats::as_polygon_atlas(dk())
# Equivalent to the default; shown explicitly:
ggplot() +
  geom_brain(atlas = poly) +
  coord_brain()

## End(Not run)
```

geom_brain

*Plot brain atlas regions***Description**

Colour brain regions by your own values. Give `geom_brain()` an atlas like `'dk()'` and a data frame, and it matches your values to the right regions and lays out the brain views for you. No data? It just draws the atlas.

Usage

```
geom_brain(
  mapping = aes(),
  data = NULL,
  atlas,
  hemi = NULL,
  view = NULL,
  position = position_brain(),
  context = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

<code>mapping</code>	Set of aesthetic mappings created by <code>[ggplot2::aes()]</code> .
<code>data</code>	A <code>data.frame</code> containing variables to map. If <code>'NULL'</code> , the atlas is plotted without user data. Group it with <code>[dplyr::group_by()]</code> to facet.
<code>atlas</code>	A <code>'ggseg_atlas'</code> object (e.g. <code>'dk()'</code> , <code>'aseg()'</code> , <code>'tracula()'</code>).
<code>hemi</code>	Character vector of hemispheres to include (e.g. <code>"left"</code> , <code>"right"</code>). Defaults to all hemispheres in the atlas.
<code>view</code>	Character vector of views to include, as recorded in the atlas data. For cortical atlases: <code>"lateral"</code> , <code>"medial"</code> . For subcortical/tract atlases: slice identifiers like <code>"axial_3"</code> . Defaults to all views.
<code>position</code>	Position adjustment, either as a string or the result of a call to <code>[position_brain()]</code> .
<code>context</code>	Keep the rest of the brain as a soft grey backdrop (<code>'TRUE'</code> , the default), or show only the regions you're plotting (<code>'FALSE'</code>).
<code>show.legend</code>	Logical. Should this layer be included in the legends?
<code>inherit.aes</code>	Logical. If <code>'FALSE'</code> , overrides the default aesthetics rather than combining with them.
<code>...</code>	Additional arguments passed to <code>[ggplot2::geom_polygon()]</code> .

Value

A list of ggplot2 layer and coord objects.

GeomBrain ggproto

‘GeomBrain’ is a [ggplot2::Geom] ggproto object that handles rendering of brain atlas polygons. It is used internally by [geom_brain()] and should not typically be called directly.

Examples

```
library(ggplot2)

ggplot() +
  geom_brain(atlas = dk())
```

position_brain	<i>Arrange brain atlas views</i>
----------------	----------------------------------

Description

Controls how an atlas’s hemispheres and views are arranged in the plot – side by side, stacked, or in a grid – and can zoom each view in on the regions you care about. Pass the result to the ‘position’ argument of [geom_brain()] (or [annotate_brain()]).

Usage

```
position_brain(
  position = "horizontal",
  nrow = NULL,
  ncol = NULL,
  views = NULL,
  zoom = NULL,
  zoom_pad = 0.05
)
```

Arguments

position	Formula describing the rows ~ columns organisation for cortical atlases (e.g., ‘hemi ~ view’). For subcortical/tract atlases, can be "horizontal", "vertical", or a formula with ‘type ~ .’ where type is extracted from view names like "axial_1" -> "axial".
nrow	Number of rows for grid layout. If NULL (default), calculated automatically. Only used for subcortical/tract atlases when position is not a formula.
ncol	Number of columns for grid layout. If NULL (default), calculated automatically. Only used for subcortical/tract atlases when position is not a formula.

views	Character vector specifying which views to include and their order. If NULL (default), all views are included in their original order. Only applies to subcortical/tract atlases.
zoom	Controls per-view zoom. 'NULL'/'FALSE' (default) draws each view at full extent. 'TRUE' zooms each view onto its focus regions; a character vector names the focus regions explicitly.
zoom_pad	Fractional padding added around the focus window when 'zoom' is active. Defaults to '0.05' (5%).

Value

A layout specification to hand to `[geom_brain()]`'s 'position' argument.

Examples

```
library(ggplot2)

# Cortical atlas with formula
ggplot() +
  geom_brain(
    atlas = dk(), aes(fill = region),
    position = position_brain(. ~ view + hemi),
    show.legend = FALSE
  )

ggplot() +
  geom_brain(
    atlas = dk(), aes(fill = region),
    position = position_brain(view ~ hemi),
    show.legend = FALSE
  )

ggplot() +
  geom_brain(
    atlas = aseg(), aes(fill = region),
    position = position_brain(nrow = 2)
  )

ggplot() +
  geom_brain(
    atlas = aseg(), aes(fill = region),
    position = position_brain(
      views = c("sagittal", "axial_3", "coronal_2"),
      nrow = 1
    )
  )

ggplot() +
  geom_brain(
    atlas = aseg(), aes(fill = region),
    position = position_brain(type ~ .)
  )
```

reposition_brain	<i>Reposition brain slices</i>
------------------	--------------------------------

Description

Repositions pre-joined sf atlas data (i.e. data and atlas already joined to a single sf data frame) for control over final plot layout. For even more detailed control, convert the "hemi" and "view" columns into factors ordered by wanted order of appearance.

Usage

```
reposition_brain(  
  data,  
  position = "horizontal",  
  nrow = NULL,  
  ncol = NULL,  
  views = NULL  
)
```

Arguments

data	sf-data.frame of joined brain atlas and data
position	Position formula for slices. For cortical atlases, use formulas like 'hemi ~ view'. For subcortical/tract atlases, use "horizontal", "vertical", or 'type ~ .' for type-based layout.
nrow	Number of rows for grid layout (subcortical/tract only)
ncol	Number of columns for grid layout (subcortical/tract only)
views	Character vector specifying view order (subcortical/tract only)

Details

This is the sf layout helper. It requires the 'sf' package (an optional dependency); for the sf-free default, build a layout with [position_brain()] and pass it to [geom_brain()].

Value

sf-data.frame with re-positioned slices

Examples

```
reposition_brain(dk(), hemi ~ view)  
reposition_brain(dk(), view ~ hemi)  
reposition_brain(dk(), hemi + view ~ .)  
reposition_brain(dk(), . ~ hemi + view)
```

```
reposition_brain(aseg(), nrow = 2)
```

```
reposition_brain(aseg(), views = c("sagittal", "axial_3"))
```

scale_brain

Colour and fill scales from brain atlas palettes

Description

‘r lifecycle::badge("deprecated")‘

Atlas palettes are now applied automatically by [geom_brain()]. Use [scale_fill_brain_manual()] for custom palettes.

Usage

```
scale_brain(
  name = "dk",
  na.value = "grey",
  ...,
  aesthetics = c("fill", "colour", "color")
)

scale_colour_brain(name = "dk", na.value = "grey", ...)

scale_color_brain(name = "dk", na.value = "grey", ...)

scale_fill_brain(name = "dk", na.value = "grey", ...)
```

Arguments

name	String name of the atlas palette (e.g. "dk", "aseg").
na.value	Colour for 'NA' entries (default: "grey").
...	Additional arguments passed to [ggseg.formats::atlas_palette()].
aesthetics	Which aesthetic to scale: "fill", "colour", or "color".

Value

A ggplot2 scale object.

Examples

```
## Not run:
library(ggplot2)
ggplot() +
  geom_brain(atlas = dk(), aes(fill = region), show.legend = FALSE) +
  scale_brain("dk")

## End(Not run)
```

scale_brain_manual *Manual colour and fill scales for brain plots*

Description

Apply a custom named colour palette to brain atlas plots. Use this when you want to override the atlas default colours with your own colour mapping.

Usage

```
scale_brain_manual(  
  palette,  
  na.value = "grey",  
  ...,  
  aesthetics = c("fill", "colour", "color")  
)
```

```
scale_colour_brain_manual(...)
```

```
scale_color_brain_manual(...)
```

```
scale_fill_brain_manual(...)
```

Arguments

palette	Named character vector mapping region names to colours.
na.value	Colour for 'NA' entries (default: "grey").
...	Additional arguments (unused).
aesthetics	Which aesthetic to scale: "fill", "colour", or "color".

Value

A ggplot2 scale object.

Examples

```
library(ggplot2)  
  
pal <- c("insula" = "red", "precentral" = "blue")  
ggplot() +  
  geom_brain(atlas = dk(), aes(fill = region), show.legend = FALSE) +  
  scale_fill_brain_manual(palette = pal)
```

 scale_brain2

Deprecated scale functions

Description

```
'r lifecycle::badge("deprecated")'
```

These functions have been renamed for clarity: - 'scale_brain2()' -> [scale_brain_manual()] - 'scale_fill_brain2()' -> [scale_fill_brain_manual()] - 'scale_colour_brain2()' -> [scale_colour_brain_manual()] - 'scale_color_brain2()' -> [scale_color_brain_manual()]

Usage

```
scale_brain2(...)
```

```
scale_colour_brain2(...)
```

```
scale_color_brain2(...)
```

```
scale_fill_brain2(...)
```

Arguments

```
... Arguments passed to the replacement function.
```

Value

A ggplot2 scale object.

Examples

```
pal <- c("transverse temporal" = "#FF0000", "insula" = "#00FF00")
suppressWarnings(scale_fill_brain_manual(palette = pal))
```

 scale_continuous_brain *Axis and label scales for brain atlas plots*

Description

Add axis labels and tick labels corresponding to brain atlas regions. These scales add hemisphere or view labels to the x and y axes based on the atlas layout.

Usage

```
scale_continuous_brain(  
  atlas = dk(),  
  position = "dispersed",  
  aesthetics = c("y", "x")  
)  
  
scale_x_brain(...)  
  
scale_y_brain(...)  
  
scale_labs_brain(atlas = dk(), position = "dispersed", aesthetics = "labs")
```

Arguments

atlas	A 'ggseg_atlas' object or data.frame containing atlas data.
position	Layout style: "dispersed" (default) or "stacked".
aesthetics	Which axis to scale: "x", "y", or "labs".
...	Additional arguments passed to [adapt_scales()].

Value

A ggplot2 scale or labs object.

Examples

```
library(ggplot2)  
  
ggplot() +  
  geom_brain(atlas = dk()) +  
  scale_x_brain() +  
  scale_y_brain() +  
  scale_labs_brain()
```

theme_brain

Themes for brain atlas plots

Description

A set of ggplot2 themes designed for brain atlas visualisations. All themes remove axis ticks and grid lines for a clean presentation.

Usage

```
theme_brain(text.size = 12, text.family = "mono")

theme_darkbrain(text.size = 12, text.family = "mono")

theme_custombrain(
  plot.background = "white",
  text.colour = "darkgrey",
  text.size = 12,
  text.family = "mono"
)

theme_brain2(
  plot.background = "white",
  text.colour = "darkgrey",
  text.size = 12,
  text.family = "mono"
)
```

Arguments

<code>text.size</code>	Text size in points (default: '12').
<code>text.family</code>	Font family (default: "mono").
<code>plot.background</code>	Background fill colour ('theme_custombrain' and 'theme_brain2' only).
<code>text.colour</code>	Text colour ('theme_custombrain' and 'theme_brain2' only).

Details

'theme_brain' Default theme. Transparent background, no axes, no grid.

'theme_darkbrain' Dark theme with black background and light text.

'theme_custombrain' Fully customisable background, text colour, size, and font.

'theme_brain2' Like 'theme_custombrain' but with axis text removed entirely.

Value

A [ggplot2::theme] object.

See Also

[geom_brain()], [ggplot2::theme()]

Examples

```
library(ggplot2)

p <- ggplot() +
  geom_brain(atlas = dk())
```

```
p +  
  theme_brain()
```

```
p +  
  theme_darkbrain()
```

Index

* datasets

- geom_brain, 6
- annotate_brain, 2
- brain_join, 4
- coord_brain, 5
- geom_brain, 6
- GeomBrain (geom_brain), 6
- position_brain, 7
- reposition_brain, 9
- scale_brain, 10
- scale_brain2, 12
- scale_brain2-deprecated (scale_brain2), 12
- scale_brain_manual, 11
- scale_color_brain (scale_brain), 10
- scale_color_brain2 (scale_brain2), 12
- scale_color_brain_manual (scale_brain_manual), 11
- scale_colour_brain (scale_brain), 10
- scale_colour_brain2 (scale_brain2), 12
- scale_colour_brain_manual (scale_brain_manual), 11
- scale_continuous_brain, 12
- scale_fill_brain (scale_brain), 10
- scale_fill_brain2 (scale_brain2), 12
- scale_fill_brain_manual (scale_brain_manual), 11
- scale_labs_brain (scale_continuous_brain), 12
- scale_x_brain (scale_continuous_brain), 12
- scale_y_brain (scale_continuous_brain), 12
- theme_brain, 13
- theme_brain2 (theme_brain), 13
- theme_custombrain (theme_brain), 13
- theme_darkbrain (theme_brain), 13