Package: ggseg3d (via r-universe)

June 6, 2026

Title Interactive 3D Brain Atlas Visualization

Version 2.1.1

Description Plot brain atlases as interactive 3D meshes using 'Three.js' via 'htmlwidgets', or render publication-quality static images through 'rgl' and 'rayshader'. A pipe-friendly API lets you map data onto brain regions, control camera angles, toggle region edges, overlay glass brains, and snapshot or ray-trace the result. Additional atlases are available through the 'ggsegverse' r-universe. Mowinckel & Vidal-Piñeiro (2020) <doi:10.1177/2515245920928009>.

License MIT + file LICENSE

URL https://github.com/ggsegverse/ggseg3d, https://ggsegverse.github.io/ggseg3d/

BugReports https://github.com/ggsegverse/ggseg3d/issues

Depends R (>= 4.1)

Imports cli, dplyr, ggseg.formats, htmlwidgets, knitr, lifecycle, rlang, scales, tidyr, webshot2

Suggests ggseg.meshes, png, rayshader, rgl, rmarkdown, shiny, spelling, testthat (>= 3.0.0)

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

Config/pak/sysreqs chromium cmake make libicu-dev libuv1-dev libssl-dev

Repository https://drmowinckels.r-universe.dev

Date/Publication 2026-04-22 20:09:50 UTC

RemoteUrl https://github.com/ggsegverse/ggseg3d

RemoteRef HEAD

RemoteSha d76567c8d11d3b3ba841217b9eaa73fcaeebc756

Contents

`add_glassbrain` *Add glass brain to ggseg3d plot*

Description

Adds a translucent brain surface to a ggseg3d plot for anatomical reference. Particularly useful for subcortical and tract visualizations where spatial context helps interpretation. Works with both htmlwidget (‘ggseg3d‘) and rgl (‘ggsegray‘) objects.

Usage

```
add_glassbrain(
  p,
  hemisphere = c("left", "right"),
  surface = "inflated",
  colour = "#CCCCCC",
  opacity = 0.3,
  brain_meshes = NULL
)
```

Arguments

p	A ‘ggseg3d‘ widget or ‘ggsegray‘ rgl object.
hemisphere	Character vector. Hemispheres to add: "left", "right", or both.
surface	Character. Surface type. Defaults to ‘"inflated"‘, which is supplied by ‘ggseg.formats‘ and requires no additional dependency. Other surfaces (‘"pial"‘, ‘"white"‘, etc.) are drawn from ‘ggseg.meshes‘.
colour	Character. Colour for the glass brain surface (hex or named).
opacity	Numeric. Transparency of the glass brain (0-1).
brain_meshes	Optional user-supplied brain meshes. See [ggseg.formats::get_brain_mesh()] for format details.

Value

The input object (modified), for piping.

Examples

```
ggseg3d(atlas = aseg()) |>
  add_glassbrain("left", opacity = 0.2)

## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(atlas = aseg()) |>
  add_glassbrain(opacity = 0.15) |>
  pan_camera("right lateral")

## End(Not run)
```

---

ggseg3d *Plot 3D brain parcellations*

Description

‘ggseg3d‘ creates and returns an interactive Three.js brain mesh visualization. Dispatches to atlas-type-specific methods via [prepare_brain_meshes()].

Usage

```
ggseg3d(
  .data = NULL,
  atlas = dk(),
  label_by = "region",
  text_by = NULL,
  colour_by = "colour",
  palette = NULL,
  na_colour = "darkgrey",
```

```
  na_alpha = 1,
  ...,
  label = deprecated(),
  text = deprecated(),
  colour = deprecated()
)
```

Arguments

`.data`	A data.frame to use for plot aesthetics. Must include a column called "region" corresponding to regions.
`atlas`	A ‘ggseg_atlas‘ object containing 3D vertex mappings, or a string naming an atlas function (deprecated).
`label_by`	String. Column name used as hover label for each region.
`text_by`	String. Column name for extra hover text shown below the region label.
`colour_by`	String. Column name mapped to mesh colours.
`palette`	String. Vector of colour names or HEX colours. Can also be a named numeric vector, with colours as names, and breakpoint for that colour as the value
`na_colour`	String. Either name, hex of RGB for colour of NA in colour.
`na_alpha`	Numeric. A number between 0 and 1 to control transparency of NA-regions.
`...`	Type-specific arguments passed to the atlas method. See section **Type-specific arguments** below.
`label`, `text`, `colour`	‘r lifecycle::badge("deprecated")‘ Use ‘label_by‘, ‘text_by‘, and ‘colour_by‘ instead.

Value

an htmlwidget object for interactive 3D brain visualization

Type-specific arguments

Cortical atlases (‘cortical_atlas‘):

‘surface‘ Surface type: ‘"LCBC"‘ (default, alias for inflated), ‘"inflated"‘, ‘"semi-inflated"‘, ‘"white"‘, ‘"pial"‘.

‘hemisphere‘ Character vector of hemispheres: ‘"right"‘, ‘"left"‘.

‘edge_by‘ Column name for region boundary edges.

‘brain_meshes‘ Custom brain mesh data.

Tract atlases (‘tract_atlas‘):

‘tract_color‘ ‘"palette"‘ (default) or ‘"orientation"‘ (direction-based RGB).

‘tube_radius‘ Tube radius (numeric, default 5).

‘tube_segments‘ Tube segment count (integer, default 8).

Author(s)

Athanasia Mowinckel and Didac Piñeiro

See Also

[pan_camera()] for camera position, [set_background()] for background colour, [set_legend()] for legend visibility

Examples

```
ggseg3d()
ggseg3d(hemisphere = "left") |> pan_camera("left lateral")
ggseg3d() |> set_legend(FALSE)
ggseg3d() |> set_background("black")
```

---

ggseg3d-shiny *Shiny bindings for ggseg3d*

Description

Output and render functions for using ggseg3d within Shiny applications and interactive R Markdown documents.

Usage

```
ggseg3dOutput(outputId, width = "100%", height = "400px")

renderGgseg3d(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	output variable to read from
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a ggseg3d
`env`	The environment in which to evaluate `expr`.
`quoted`	Is `expr` a quoted expression (with `quote()`)? This is useful if you want to save an expression in a variable.

Value

‘ggseg3dOutput‘ returns an HTML widget output element for use in a Shiny UI. ‘renderGgseg3d‘ returns a render function for use in a Shiny server.

Examples

```
library(shiny)
ui <- fluidPage(ggseg3dOutput("brain"))
server <- function(input, output) {
  output$brain <- renderGgseg3d(ggseg3d())
}
```

---

ggsegray *Render brain atlas with rgl*

Description

Creates an rgl 3D scene from a brain atlas. Uses the same atlas preparation pipeline as [ggseg3d()] but outputs to rgl instead of htmlwidgets. The resulting scene can be piped into [pan_camera()], [add_glassbrain()], and [set_background()], then rendered with rayshader's ‘render_highquality()‘ or captured with ‘rgl::snapshot3d()‘.

Usage

```
ggsegray(
  .data = NULL,
  atlas = dk(),
  label_by = "region",
  text_by = NULL,
  colour_by = "colour",
  palette = NULL,
  na_colour = "darkgrey",
  na_alpha = 1,
  material = list(),
  ...,
  label = deprecated(),
  text = deprecated(),
  colour = deprecated()
)
```

Arguments

.data	A data.frame to use for plot aesthetics. Must include a column called "region" corresponding to regions.
atlas	A ‘ggseg_atlas‘ object containing 3D vertex mappings, or a string naming an atlas function (deprecated).
label_by	String. Column name used as hover label for each region.
text_by	String. Column name for extra hover text shown below the region label.
colour_by	String. Column name mapped to mesh colours.

`palette` String. Vector of colour names or HEX colours. Can also be a named numeric vector, with colours as names, and breakpoint for that colour as the value

`na_colour` String. Either name, hex of RGB for colour of NA in colour.

`na_alpha` Numeric. A number between 0 and 1 to control transparency of NA-regions.

`material` Named list of rgl material properties passed to [rgl::tmesh3d()]. Controls how the mesh surface is shaded.

`...` Type-specific arguments passed to the atlas method. See section **Type-specific arguments** below.

`label`, `text`, `colour`
'r lifecycle::badge("deprecated")' Use 'label_by', 'text_by', and 'colour_by' instead.

Value

An object of class 'ggsegray' (invisibly), which wraps the rgl device ID. Pipe into [pan_camera()], [add_glassbrain()], or [set_background()] to modify the scene.

Material properties

Useful material list entries:

'specular' '"black"' (matte) or '"white"' (glossy).

'shininess' Specular exponent. Higher = tighter highlights.

'lit' 'FALSE' disables lighting.

'alpha' Transparency, 0 (invisible) to 1 (opaque).

'smooth' 'TRUE' for Gouraud shading, 'FALSE' for flat.

See [rgl::material3d()] for the full list.

Type-specific arguments

Cortical atlases ('cortical_atlas'):

'surface' Surface type: '"LCBC"' (default, alias for inflated), '"inflated"', '"semi-inflated"', '"white"', '"pial"'.

'hemisphere' Character vector of hemispheres: '"right"', '"left"'.

'edge_by' Column name for region boundary edges.

'brain_meshes' Custom brain mesh data.

Tract atlases ('tract_atlas'):

'tract_color' '"palette"' (default) or '"orientation"' (direction-based RGB).

'tube_radius' Tube radius (numeric, default 5).

'tube_segments' Tube segment count (integer, default 8).

Examples

```
## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(hemisphere = "left") |>
  pan_camera("left lateral")

ggsegray(atlas = aseg()) |>
  add_glassbrain(opacity = 0.15) |>
  pan_camera("right lateral") |>
  set_background("black")

## End(Not run)
```

`pan_camera` *Pan camera position of ggseg3d plot*

Description

Sets the camera position for a ggseg3d widget or ggsegray rgl scene to standard anatomical views or custom positions.

Usage

```
pan_camera(p, camera)
```

Arguments

`p`	A ‘ggseg3d‘ widget or ‘ggsegray‘ rgl object.
`camera`	string, list, or numeric vector. Camera position preset name, custom eye position list, or ‘c(x, y, z)‘ for rgl.

Available camera presets:

- ‘left lateral‘ or ‘left_lateral‘
- ‘left medial‘ or ‘left_medial‘
- ‘right lateral‘ or ‘right_lateral‘
- ‘right medial‘ or ‘right_medial‘
- ‘left superior‘ or ‘left_superior‘
- ‘right superior‘ or ‘right_superior‘
- ‘left inferior‘ or ‘left_inferior‘
- ‘right inferior‘ or ‘right_inferior‘
- ‘left anterior‘ or ‘left_anterior‘
- ‘right anterior‘ or ‘right_anterior‘
- ‘left posterior‘ or ‘left_posterior‘
- ‘right posterior‘ or ‘right_posterior‘

Value

The input object (modified), for piping.

Examples

```
ggseg3d() |> pan_camera("right lateral")
## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(atlas = dk(), hemisphere = "left") |>
  pan_camera("left lateral")

## End(Not run)
```

`resolve_brain_mesh` *Resolve brain surface mesh*

Description

Resolves and prepares a brain surface mesh for rendering. Delegates to [ggseg.formats::get_brain_mesh()] for inflated surfaces and to [ggseg.meshes::get_cortical_mesh()] for pial, white, semi-inflated, and other surfaces. Returned meshes use the shared anatomical axis convention (‘x‘ = left-right, ‘y‘ = anterior-posterior, ‘z‘ = superior-inferior) with ‘lh‘ positioned at ‘x <= 0‘ and ‘rh‘ at ‘x >= 0‘, medial edges meeting at the midline (‘x = 0‘).

Usage

```
resolve_brain_mesh(
  hemisphere = c("lh", "rh"),
  surface = c("inflated", "semi-inflated", "white", "pial", "sphere", "smoothwm", "orig"),
  brain_meshes = NULL
)
```

Arguments

`hemisphere`	‘"lh"‘ or ‘"rh"‘
`surface`	Surface type: ‘"inflated"‘, ‘"semi-inflated"‘, ‘"white"‘, ‘"pial"‘, ‘"sphere"‘, ‘"smoothwm"‘, ‘"orig"‘
`brain_meshes`	Optional user-supplied mesh data. Passed through to [ggseg.formats::get_brain_mesh()] for format details.

Value

list with vertices (data.frame with x, y, z) and faces (data.frame with i, j, k), or NULL if mesh not found

Examples

```
mesh <- resolve_brain_mesh("lh", "inflated")
str(mesh, max.level = 1)
```

`set_background` *Set background color of ggseg3d plot*

Description

Changes the background color of a ggseg3d widget or ggsegray rgl scene.

Usage

```
set_background(p, colour = "#ffffff")
```

Arguments

`p`	A ‘ggseg3d‘ widget or ‘ggsegray‘ rgl object.
`colour`	string. Background color (hex or named color)

Value

The input object (modified), for piping.

Examples

```
ggseg3d() |> set_background("black")
## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(atlas = dk()) |> set_background("black")

## End(Not run)
```

`set_dimensions` *Set widget dimensions*

Description

Changes the width and height of a ggseg3d widget.

Usage

```
set_dimensions(p, width = NULL, height = NULL)
```

Arguments

p	ggseg3d widget object
width	numeric. Widget width in pixels (NULL for default)
height	numeric. Widget height in pixels (NULL for default)

Value

ggseg3d widget object with updated dimensions

Examples

```
ggseg3d() |>
  set_dimensions(width = 800, height = 600)
```

---

set_edges *Set region boundary edges*

Description

Adds coloured outlines around brain regions. This is useful for highlighting region boundaries in figures. Works with both htmlwidget (‘ggseg3d‘) and rgl (‘ggsegray‘) objects. For rgl, edges must have been computed at creation time via ‘edge_by‘.

Usage

```
set_edges(p, colour = "black", width = 1)
```

Arguments

p	A ‘ggseg3d‘ widget or ‘ggsegray‘ rgl object.
colour	string. Edge colour (hex or named color). Set to NULL to hide edges.
width	numeric. Width of edge lines (default: 1). Note: line width > 1 may not render on all systems due to WebGL limitations.

Value

The input object (modified), for piping.

Lifecycle

‘r lifecycle::badge("experimental")‘

Examples

```
ggseg3d(hemisphere = "left", edge_by = "region") |>
  set_edges("black") |>
  pan_camera("left lateral")
## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(hemisphere = "left", edge_by = "region") |>
  set_edges("red", width = 2) |>
  pan_camera("left lateral")

## End(Not run)
```

`set_flat_shading` *Enable flat shading for ggseg3d plot*

Description

Disables lighting effects to show colors exactly as specified. Useful for screenshots where accurate color reproduction is needed, such as atlas creation pipelines that extract contours from images.

Usage

```
set_flat_shading(p, flat = TRUE)
```

Arguments

`p`	ggseg3d widget object
`flat`	logical. Enable flat shading (default: TRUE)

Value

ggseg3d widget object with updated shading

Examples

```
ggseg3d() |>
  set_flat_shading()
```

set_legend *Set legend visibility*

Description

For htmlwidget output, toggles legend visibility. For rgl output, draws or removes the legend overlay.

Usage

```
set_legend(p, show = TRUE)
```

Arguments

p	A ggseg3d or ggsegray object
show	logical. Whether to show the legend (default: TRUE)

Value

The input object, modified

Examples

```
ggseg3d() |> set_legend(FALSE)
## Not run:
# rgl requires OpenGL; not run in check environments.
ggsegray(hemisphere = "left") |> set_legend()

## End(Not run)
```

set_orthographic *Enable orthographic camera for ggseg3d plot*

Description

Uses orthographic projection instead of perspective. This eliminates perspective distortion and ensures consistent sizing across all views.

Usage

```
set_orthographic(p, ortho = TRUE, frustum_size = 220)
```

Arguments

p	ggseg3d widget object
ortho	logical. Enable orthographic mode (default: TRUE)
frustum_size	numeric. Size of the orthographic frustum. Controls how much of the scene is visible. Default 220 works well for brain meshes. Use the same value across all views for consistent sizing.

Value

ggseg3d widget object with updated camera mode

Examples

```
ggseg3d() |>
  set_orthographic()
```

---

set_positioning *Set hemisphere positioning mode*

Description

Repositions cortical hemisphere meshes in a ggseg3d widget. Meshes are produced with anatomical positioning by default (medial edges at x = 0). Use this to centre each hemisphere at the origin for atlas-creation snapshots, or to reapply anatomical positioning after manual edits.

Usage

```
set_positioning(p, positioning = c("anatomical", "centered"))
```

Arguments

p	ggseg3d widget object
positioning	How to position hemispheres: - "anatomical": Medial surfaces adjacent at midline. Left at negative x, right at positive x. Default for displaying both hemispheres together. - "centered": Centre each hemisphere at the origin. Best for single-hemisphere snapshots where consistent sizing is needed.

Details

Only cortical hemisphere meshes are repositioned (those named ‘"<hemi> <surface>"‘, e.g. ‘"left inflated"‘). Subcortical, cerebellar and glassbrain meshes are left untouched.

Value

ggseg3d widget object with repositioned meshes

Examples

```
# View both hemispheres anatomically positioned (default)
ggseg3d(hemisphere = c("left", "right")) |>
  pan_camera("left lateral")

# Atlas creation: centered for consistent sizing
ggseg3d(hemisphere = "left") |>
  set_positioning("centered") |>
  set_orthographic() |>
  pan_camera("left lateral")
```

---

snapshot_brain *Save ggseg3d widget as image*

Description

Takes a screenshot of a ggseg3d widget and saves it as a PNG image. Requires a Chrome-based browser to be installed.

Usage

```
snapshot_brain(p, file, width = 600, height = 500, delay = 1, zoom = 2, ...)
```

Arguments

p	ggseg3d widget object
file	string. Output file path (should end in .png)
width	numeric. Image width in pixels (default: 600)
height	numeric. Image height in pixels (default: 500)
delay	numeric. Seconds to wait for widget to render before capture (default: 1)
zoom	numeric. Zoom factor for higher resolution (default: 2)
...	Additional arguments passed to webshot2::webshot

Value

The file path (invisibly)

Examples

```
## Not run:
ggseg3d() |>
  pan_camera("left lateral") |>
  snapshot_brain("brain.png")

## End(Not run)
```

updateGgseg3dBackground *Update background in Shiny*

Description

Sends a message to update the background color of a ggseg3d widget in a Shiny app.

Usage

```
updateGgseg3dBackground(session, outputId, colour)
```

Arguments

session	The Shiny session object
outputId	The output ID of the ggseg3d widget
colour	Background color (hex or named color)

Value

None, called for side effects (sends message to client)

Examples

```
## Not run:
updateGgseg3dBackground(session, "brain", "black")

## End(Not run)
```

updateGgseg3dCamera *Update camera in Shiny*

Description

Sends a message to update the camera position of a ggseg3d widget in a Shiny app.

Usage

```
updateGgseg3dCamera(session, outputId, camera)
```

Arguments

session	The Shiny session object
outputId	The output ID of the ggseg3d widget
camera	Camera position preset or custom position

Value

None, called for side effects (sends message to client)

Examples

```
## Not run:
updateGgseg3dCamera(session, "brain", "left lateral")

## End(Not run)
```

Index