

Package: ggseg3d (via r-universe)

October 14, 2026

Title Interactive 3D Brain Atlas Visualization

Version 2.1.2.9004

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 (>= 0.0.4.9010), htmlwidgets, knitr, lifecycle, rlang, scales, tidyr, webshot2

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

VignetteBuilder knitr

Config/Needs/website ggsegverse/ggseg.docs, png, rayshader

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

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

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

Date/Publication 2026-10-14 18:29:46 UTC
RemoteUrl <https://github.com/ggsegverse/ggseg3d>
RemoteRef HEAD
RemoteSha 272fd34a4c1b1bf8b340c4f20f9b92974e58f5cf

Contents

add_glassbrain	2
ggseg3d	3
ggseg3d-shiny	5
ggsegray	6
pan_camera	8
resolve_brain_mesh	10
set_background	11
set_dimensions	11
set_edges	12
set_flat_shading	13
set_legend	14
set_orthographic	14
set_positioning	15
snapshot_brain	16
updateGgseg3dBackground	17
updateGgseg3dCamera	18
Index	19

add_glassbrain	<i>Add glass brain to ggseg3d plot</i>
----------------	--

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. The default "inflated" surface ships with ggseg.formats; every other surface is drawn from ggseg.meshes.

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. "lh" and "rh" are accepted as well.
surface	Surface type: "inflated", "semi-inflated", "white", "pial", "midthickness", "sphere", "smoothwm" or "orig". ggseg3d() additionally accepts "LCBC", an alias for "inflated".
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 mesh data. Passed through to ggseg.formats::get_brain_mesh() for format details.

Value

The input object (modified), for piping.

See Also

Other brain modifiers: [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

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

<code>.data</code>	A data.frame to use for plot aesthetics. Must include a column called "region" corresponding to regions.
<code>atlas</code>	A ggseg_atlas object containing 3D vertex mappings, or a string naming an atlas function (deprecated).
<code>label_by</code>	String. Column name used as hover label for each region.
<code>text_by</code>	String. Column name for extra hover text shown below the region label.
<code>colour_by</code>	String. Column name mapped to mesh colours.
<code>palette</code>	Character vector of colour names or hex colours. May instead be a named numeric vector whose names are the colours and whose values are the breakpoints those colours sit at.
<code>na_colour</code>	String. Colour drawn for regions with no value in <code>.data</code> , as a colour name or a hex code.
<code>na_alpha</code>	Numeric between 0 and 1. Opacity of regions with no value in <code>.data</code> : 1 (default) draws them opaque in <code>na_colour</code> , lower values fade them. Cortical and cerebellar surfaces fade per vertex, subcortical structures and tracts per mesh.
<code>...</code>	Type-specific arguments passed to the atlas method. See section Type-specific arguments below.
<code>label, text, colour</code>	[Deprecated] Use <code>label_by</code> , <code>text_by</code> , and <code>colour_by</code> instead.

Value

an htmlwidget object for interactive 3D brain visualization

Type-specific arguments

Cortical atlases (`cortical_atlas`):

surface "LCBC" (default, an alias for "inflated") or any surface `resolve_brain_mesh()` accepts.

hemisphere Character vector of hemispheres: "left", "right", or their "lh"/"rh" spellings.

edge_by Column name for region boundary edges.

brain_meshes Custom brain mesh data.

Cerebellar atlases (cerebellar_atlas):

surface_opacity Opacity of the cerebellar surface mesh. Defaults to 0.3 when the atlas carries deep nuclei, so they stay visible through the surface, and 1 otherwise.

Tract atlases (tract_atlas):

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

tube_radius Tube radius (numeric, default 2).

tube_segments Tube segment count (integer, default 10).

Author(s)

Athanasia Mowinckel and Didac Piñeiro

See Also

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

Other brain plots: `ggsegray()`

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

<code>.data</code>	A data.frame to use for plot aesthetics. Must include a column called "region" corresponding to regions.
<code>atlas</code>	A ggseg_atlas object carrying 3D geometry.
<code>label_by</code>	String. Column name used as hover label for each region.
<code>text_by</code>	String. Column name for extra hover text shown below the region label.
<code>colour_by</code>	String. Column name mapped to mesh colours.
<code>palette</code>	Character vector of colour names or hex colours. May instead be a named numeric vector whose names are the colours and whose values are the breakpoints those colours sit at.
<code>na_colour</code>	String. Colour drawn for regions with no value in <code>.data</code> , as a colour name or a hex code.
<code>na_alpha</code>	Numeric between 0 and 1. Opacity of regions with no value in <code>.data</code> : 1 (default) draws them opaque in <code>na_colour</code> , lower values fade them. Cortical and cerebellar surfaces fade per vertex, subcortical structures and tracts per mesh.
<code>material</code>	Named list of rgl material properties passed to <code>rgl::tmesh3d()</code> . Controls how the mesh surface is shaded.
<code>...</code>	Type-specific arguments passed to the atlas method. See section Type-specific arguments below.
<code>label, text, colour</code>	[Deprecated] Use <code>label_by</code> , <code>text_by</code> , and <code>colour_by</code> 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 "LCBC" (default, an alias for "inflated") or any surface `resolve_brain_mesh()` accepts.

hemisphere Character vector of hemispheres: "left", "right", or their "lh"/"rh" spellings.

edge_by Column name for region boundary edges.

brain_meshes Custom brain mesh data.

Cerebellar atlases (cerebellar_atlas):

surface_opacity Opacity of the cerebellar surface mesh. Defaults to 0.3 when the atlas carries deep nuclei, so they stay visible through the surface, and 1 otherwise.

Tract atlases (tract_atlas):

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

tube_radius Tube radius (numeric, default 2).

tube_segments Tube segment count (integer, default 10).

See Also

Other brain plots: `ggseg3d()`

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.

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

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	<i>Resolve brain surface mesh</i>
--------------------	-----------------------------------

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 $x \leq 0$ and rh at $x \geq 0$, medial edges meeting at the midline ($x = 0$).

Usage

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

Arguments

hemisphere	"lh" or "rh"
surface	Surface type: "inflated", "semi-inflated", "white", "pial", "midthickness", "sphere", "smoothwm" or "orig". <code>ggseg3d()</code> additionally accepts "LCBC", an alias for "inflated".
brain_meshes	Optional user-supplied mesh data. Passed through to <code>ggseg.formats::get_brain_mesh()</code> 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	<i>Set background color of ggseg3d plot</i>
----------------	---

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.

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

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	<i>Set widget dimensions</i>
----------------	------------------------------

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

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

Examples

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

set_edges	<i>Set region boundary edges</i>
-----------	----------------------------------

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

[Experimental]

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

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	<i>Enable flat shading for ggseg3d plot</i>
------------------	---

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

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

Examples

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

set_legend	<i>Set legend visibility</i>
------------	------------------------------

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 widget or ggsegray rgl object.
show	logical. Whether to show the legend (default: TRUE)

Value

The input object, modified

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_orthographic\(\)](#), [set_positioning\(\)](#)

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	<i>Enable orthographic camera for ggseg3d plot</i>
------------------	--

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

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_positioning\(\)](#)

Examples

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

set_positioning	<i>Set hemisphere positioning mode</i>
-----------------	--

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

See Also

Other brain modifiers: [add_glassbrain\(\)](#), [pan_camera\(\)](#), [set_background\(\)](#), [set_dimensions\(\)](#), [set_edges\(\)](#), [set_flat_shading\(\)](#), [set_legend\(\)](#), [set_orthographic\(\)](#)

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

<code>session</code>	The Shiny session object
<code>outputId</code>	The output ID of the ggseg3d widget
<code>colour</code>	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	<i>Update camera in Shiny</i>
---------------------	-------------------------------

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

* brain modifiers

- add_glassbrain, 2
- pan_camera, 8
- set_background, 11
- set_dimensions, 11
- set_edges, 12
- set_flat_shading, 13
- set_legend, 14
- set_orthographic, 14
- set_positioning, 15

* brain plots

- ggseg3d, 3
- ggsegray, 6

add_glassbrain, 2
add_glassbrain(), 6, 7, 9, 11–16

ggseg.formats::get_brain_mesh(), 3, 10
ggseg.meshes::get_cortical_mesh(), 10
ggseg3d, 3
ggseg3d(), 3, 6, 8, 10
ggseg3d-shiny, 5
ggseg3dOutput (ggseg3d-shiny), 5
ggsegray, 6
ggsegray(), 5

pan_camera, 8
pan_camera(), 3, 5–7, 11–16
prepare_brain_meshes(), 3

renderGgseg3d (ggseg3d-shiny), 5
resolve_brain_mesh, 10
resolve_brain_mesh(), 5, 8
rgl::material3d(), 7
rgl::tmesh3d(), 7

set_background, 11
set_background(), 3, 5–7, 9, 12–16
set_dimensions, 11
set_dimensions(), 3, 9, 11, 13–16
set_edges, 12

set_edges(), 3, 9, 11–16
set_flat_shading, 13
set_flat_shading(), 3, 9, 11–16
set_legend, 14
set_legend(), 3, 5, 9, 11–13, 15, 16
set_orthographic, 14
set_orthographic(), 3, 9, 11–14, 16
set_positioning, 15
set_positioning(), 3, 9, 11–15
snapshot_brain, 16

updateGgseg3dBackground, 17
updateGgseg3dCamera, 18