Visualisation of Brain Statistics with R-packages *ggseg* and *ggseg3d*

Athanasia M. Mowinckel[1] **and Didac Vidal-Piñeiro**[1]

[1]**Center for Lifespan Changes in Brain and Cognition, University of Oslo, PO. box 1094 Blindern, 0317 Oslo, Norway**

Corresponding author:
Athanasia M. Mowinckel[1]

Email address: a.m.mowinckel@psykologi.uio.no

ABSTRACT

There is an increased emphasis on visualizing neuroimaging results in more intuitive ways. Common statistical tools for dissemination, such as bar charts, lack the spatial dimension that is inherent in neuroimaging data. Here we present two packages for the statistical software R, *ggseg* and *ggseg3d*, that integrate this spatial component. The *ggseg* and *ggseg3d* packages visualize pre-defined brain segmentations as both 2D polygons and 3D meshes, respectively. Both packages are integrated with other well-established R-packages, allowing great flexibility. In this tutorial, we present the main data and functions in the *ggseg* and *ggseg3d* packages for brain atlas visualization. The main highlighted functions are able to display brain segmentation plots in R. Further, the accompanying *ggsegExtra*-package includes a wider collection of atlases, and is intended for community-based efforts to develop more compatible atlases to *ggseg* and *ggseg3d*. Overall, the *ggseg*-packages facilitate parcellation-based visualizations in R, improve and ease the dissemination of the results, and increase the efficiency of the workflows.

1 INTRODUCTION

Visualization is increasingly important for accurate guidance and interpretation of neuroimaging results, as current research is able to generate a high amount of data and outcomes. For Magnetic Resonance Imaging (MRI), neuroimaging software provides whole-brain information by using many small units of space (>100.000). Nonetheless, this data is often grouped and summarized into a limited number of regions using predefined brain parcellation atlases. Brain parcellations segment the brain into a finite set of meaningful neurobiological components, which reflect one or more brain features either based on structural or connectivity properties (Eickhoff et al. [2018]). The use of brain atlases is widespread as these facilitate interpretation and minimize the amount of data, hence reducing problems with multiple comparisons. This enables replicability and data sharing in otherwise computationally expensive analyses, which are often performed in specialized software environments such as R (R Core Team [2019]).

MRI data provides good spatial resolution and thus an optimal representation has to respect spatial relationships across regions. Results from brain atlas analyses are most meaningfully visualized when projected onto a representation of the brain, thus it is desirable that any visual representation takes this relation into account. The projection of data onto brain representations provides clear points of reference - especially when the reader is unfamiliar with the atlas - eases readability, guides interpretation, and conveys the spatial patterns of the data. Adopting the grammar of graphics implemented in *ggplot2* (Wickham [2016]), one can plot neuroimaging data directly in R with several tools such as ggBrain (Fisher [2019]) and ggneuro (Muschelli [2017]; see neuroconductor [2018] for curated neuroimaging packages for R). Yet these tools display whole-brain image files and are not well-suited for representing brain atlas data.

In this tutorial, we introduce two packages for visualizing brain atlas data in R. The *ggseg* and *ggseg3d* – plus the complimentary *ggsegExtra* – packages include pre-compiled data sets for different brain atlases that allow for 2D and 3D visualization. The two-dimensional functionality in *ggseg* is based on polygons and *ggplot2*-based grammar of graphics (Wickham [2016]), while the 3D functionality in *ggseg3d* is

Table 1. Table of currently available atlases in the ggseg, ggseg3d, and the ggsegExtra R-packages. Polygon and mesh refer to 2D and 3D brain atlas representations, respectively

Title	Item	Mesh	Polygon	Citation
Automated probabilistic reconstruction of white-matter pathways	tracula	None	ggsegExtra	Yendiki et al. [2011]
Desikan-Killiany Cortical Atlas	dkt	ggseg3d	ggseg	Desikan et al. [2006]
Desterieux cortical parcellations	desterieux	ggsegExtra	None	Destrieux et al. [2010]
Freesurfer ASEG with posterior-anterior hippocampus	hcpa	ggsegExtra	None	
Freesurfer automatic subcortical segmentation of a brain volume	aseg	ggseg3d	ggseg	Fischl et al. [2002]
Genetic topography of brain area morphology	chenAr	None	ggsegExtra	Chen et al. [2013]
Genetic topography of brain thickness morphology	chenTh	None	ggsegExtra	Chen et al. [2013]
Harvard-Oxford Cortical atlas	hoCort	None	ggsegExtra	Makris et al. [2006]
HPC - Multi-modal parcellation of human cerebral cortex	glasser	None	ggsegExtra	Glasser et al. [2016]
ICBM white matter parcellation	icbm	ggsegExtra	None	Mori et al. [2005]
JHU parcellation	jhu	None	ggsegExtra	Hua et al. [2008]
Local-Global 17 Parcellation of the Human Cerebral Cortex	schaefer17	ggsegExtra	None	Schaefer et al. [2017]
Local-Global 7 Parcellation of the Human Cerebral Cortex	schaefer7	ggsegExtra	None	Schaefer et al. [2017]
Parcellation from JHU	jhu	ggsegExtra	None	Hua et al. [2008]
Parcellation from the Human Connectome Project	glasser	ggsegExtra	None	Glasser et al. [2016]
White matter tract parcellations	tracula	ggsegExtra	None	Yendiki et al. [2011]
Yeo 17 Resting-state Cortical Parcellations	yeo17	ggsegExtra	ggsegExtra	Yeo et al. [2011]
Yeo 7 Resting-state Cortical Parcellations	yeo7	ggsegExtra	ggsegExtra	Yeo et al. [2011]

based on tri-surface mesh plots and *plotly* (Sievert [2018]).

Both packages present compiled data sets, tailored functions that allow brain data integration and plotting, and other minor features such as custom colour palettes. The data featured in the packages are derived from two well-known parcellations: the Desikan-Killany cortical atlas (DKT; Desikan et al. [2006]), which covers the cortical surface of the brain, and the Automatic Segmentation of Subcortical Structures (aseg; Fischl et al. [2002]), which covers the subcortical structures. Both atlases are implemented in several neuroimaging softwares, such as FreeSurfer (Fischl et al. [1999], Dale et al. [1999], Fischl and Dale [2000]), and are commonly used in relation to developmental changes, disease biomarkers, genomic data, and cognition (Amlien et al. [2019], Walhovd et al. [2005], Pizzagalli et al. [2009]). The *ggsegExtra* package contains a collection of precompiled atlases (currently 16 additional atlases) and it is frequently updated. A summary of all available atlases compatible with the *ggseg*-packages as of March 2020 can be found in Table 1.

2 TUTORIAL

This tutorial will introduce the *ggseg*, *ggseg3d*, and *ggsegExtra* packages and familiarize the reader with the main functions and the general use of the packages. The tutorial will focus on the two main functions: `ggseg()` for plotting 2D polygons and `ggseg3d()` for plotting 3D brains based on tri-surface mesh plots.

2.1 Plotting polygon data (ggplot2)

ggseg is the main function for plotting 2D data. By default, the function automatically plots the DKT atlas (see Figure 1). The `ggseg()`-function is a wrapper for `geom_polygon()` from *ggplot2*, and it can be built upon and combined like any *ggplot*-object. The image plot consists of a simple brain representation containing no extra information. Hence, *ggseg* plots can be easily complemented with any of the available *ggplot2* features and options. We recommend users to get familiarized with *ggplot2* (Wickham [2016]). The package is currently only available through github, but we expect to submit the *ggseg*-package to The Comprehensive R Archive Network (Hornik [2012]) in 2020.

```
# remotes::install_github("LCBC-UiO/ggseg")
library(ggseg)

# package combining plots easily
library(patchwork)
```

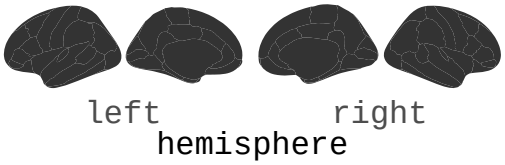

Figure 1. By default `ggseg()` will plot the dkt atlas in grey shaded polygons.

```
# Figure 1
ggseg()
```

In addition to the standard options for *ggplot2* polygon geoms, the function also has several options for plotting the main brain representations. These options are atlas-specific. For cortical atlases, such as the `dkt`, one can stack the hemispheres, display only the medial or lateral side, choose either one or both hemispheres, or any combination of hemisphere and view (see Figure 2 for examples). For subcortical atlases, such as the `aseg` atlas, the options are more limited but one can often choose between axial, sagittal, and coronal views.

```
# A: dkt dark theme
p1 <- ggseg(position = "stacked") +
  theme_dark() +
  labs(title="dkt", subtitle = "dark theme")

# B: dkt medial view
p2 <- ggseg(view = "medial") +
  labs(title = "dkt", subtitle = "medial view")

# C: dkt left medial alone
p3 <- ggseg(view = "medial",
            hemisphere = "left") +
  labs(title="dkt", subtitle = "medial left side")

# D: dkt classic theme
p4 <- ggseg(position = "stacked") +
  theme_classic() +
  labs(title = "dkt", subtitle = "classic theme")


# E: dkt left hemisphere
p5 <- ggseg(hemisphere = "left") +
  labs(title = "dkt", subtitle = "left hemisphere")

# F: aseg default theme
p6 <- ggseg(atlas = aseg, view = "axial") +
  labs(title = "aseg", subtitle = "axial view")

# Using patchwork to combine all plots into one
(p1 | p2 | p3) / (p4 | p5 | p6) +
  plot_annotation(tag_levels = 'A')
```

2.1.1 Using own data with fill and colour

`ggseg()` accepts any argument you can supply to `geom_polygon()`, and therefore is easy to work with for those familiar with *ggplot2* functionality. Standard arguments like `fill` that floods the segments with a colour, or `colour` that colours the edges around the segments are typical arguments to provide to

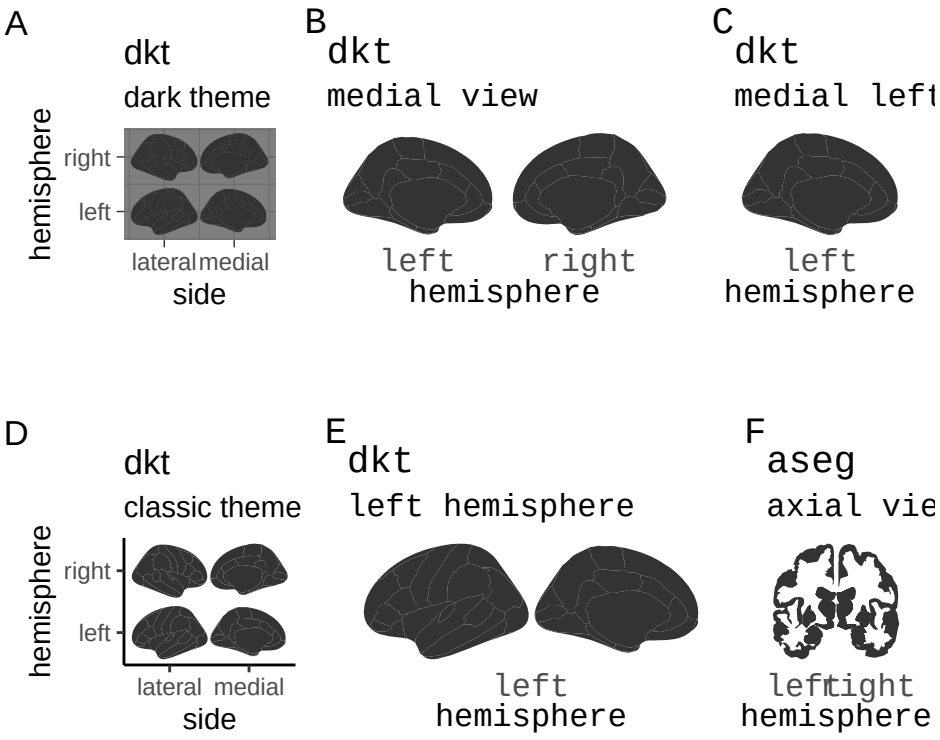

Figure 2. *ggseg* plots can be used with any *ggplot2* feature such as standard scales and themes. For cortical atlases, one can supply special *ggseg* options to determine, for example, hemisphere, view, or position. **A:** `dkt` atlas, stacked with dark theme ; **B:** `dkt` with medial view only; **C:** `dkt` atlas with only left medial display; **D:** `dkt` atlas, stacked, with classic theme; **E:** `dkt` atlas with left hemisphere only; **F:** axial view of `aseg` atlas

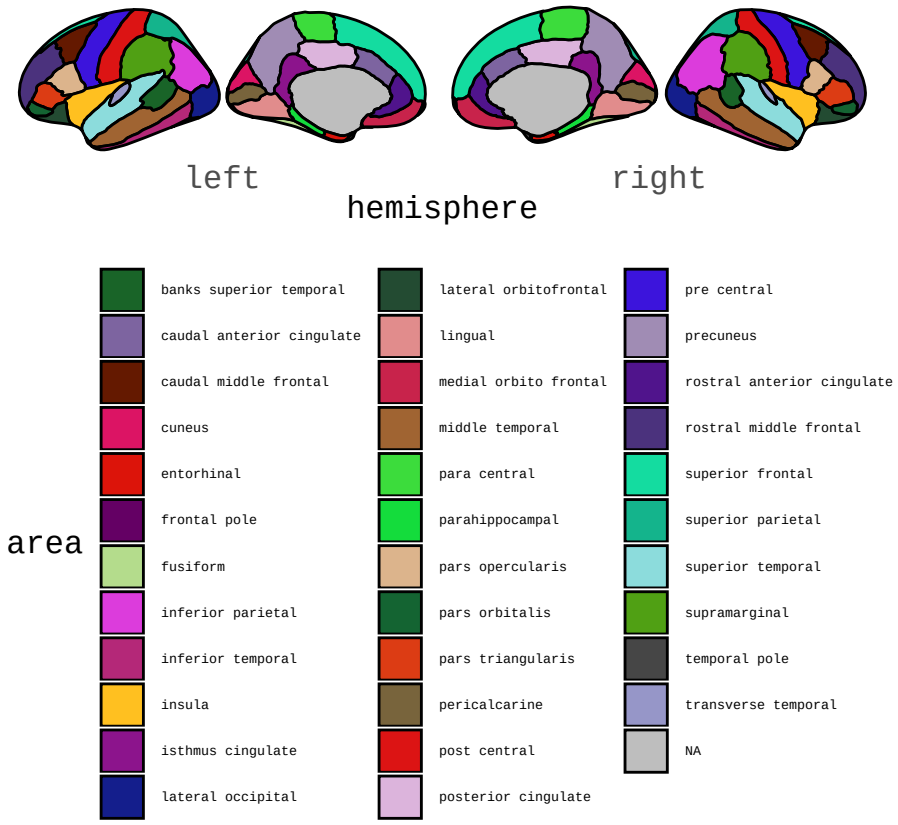

Figure 3. Supplying 'area' to the fill option in `ggseg()`, will use the column 'area' from the accompanying dataset to create a discrete colour palette over the segments in the atlas. The `dkt` atlas default palette corresponds to the FreeSurferColorLut scheme.

the function either as a single setting value or within the *ggplot2* mapping function `aes`. To use color palettes corresponding to those used in the original neuroimaging softwares one can use atlas-specific 'brain' palette scales (Figure 3).

```
# Figure 3
ggseg(mapping = aes(fill = area), colour="black") +
  scale_fill_brain("dkt") +
  theme(legend.justification = c(1,0),
        legend.position = "bottom",
        legend.text = element_text(size = 5)) +
  guides(fill = guide_legend(ncol = 3))
```

Most users will use `ggseg()` to display - using a color scale - some descriptive or inferential statistics, such as mean thickness or brain-cognition relationships across the different brain regions. Yet, before projecting the statistics onto the segments, one should explore the structure of the atlas data sets. The atlas data set structure will help users understand what incoming statistical data needs to look like. Note that each atlas corresponds to a unique data set. All data sets have a similar structure and contain key information regarding the atlas, the region names, and the coordinates for the segment polygons.

```
# Look at the top 5 rows of the dkt dataset
head(dkt, 5)
```

In any atlas, the column 'label' is particularly useful for combining the data of interest with the *ggseg*-polygons. The column 'label' contains the label (region) names as in the original neuroimaging software. For example, the DKT atlas label column matches the region names from Freesurfer statistics

table outputs. Yet, the data in *ggseg* is in a long format - that is each region has a single row - and any data of interest needs to be in this same format. Often data sets are organized in wide format, in which subjects are represented by rows and each different data variable is represented in a separate column, and thus need to be rearranged to work with ggseg. To ease to conversion between Freesurfer files and ggseg compatible files, we have functions to aid import to R. There is an extended vignette in the online documentation of the various options, here we assume that that the data has been compiled into a file using Freesurfer's `aparcstats2table` or `asegstats2table` that combine parcellation metrics for all wanted subjects into a single file. This file also contains extra metrics like intracranial volume and total grey matter. These do not map onto ggseg, and as such, one should make sure to remove them from the subsequent data.

```
file_path <- "data/aparc.volume.table"

someData <- read_freesurfer_table(file_path, measure = "volume") %>%
  # remove rows in the data that do not conain the strings "rh" or "lh"
  dplyr::filter(grepl("rh|lh", label))
someData
## # A tibble: 34 x 3
##    subject label                        volume
##    <fct>   <chr>                         <dbl>
##  1 bert    rh_bankssts                    1969
##  2 bert    rh_caudalanteriorcingulate     2280
##  3 bert    rh_caudalmiddlefrontal         5390
##  4 bert    rh_cuneus                      2998
##  5 bert    rh_entorhinal                  1735
##  6 bert    rh_fusiform                    8144
##  7 bert    rh_inferiorparietal           14876
##  8 bert    rh_inferiortemporal           11016
##  9 bert    rh_isthmuscingulate            1983
## 10 bert    rh_lateraloccipital           12729
## # ... with 24 more rows
```

A second option is to read in atlas files directly from the entire subject directory, using regular expression to define the atlas wanted. If we inspect a single subjects stats directory of Freesurfer (here on default installation on macOS), we can see several parcellations to choose from. The default aparc has a unique string ending with `aparc.stats` which we will use to read in all subject stats for that particular atlas. This data has more columns, as it contains several metrics that can be plotted.

```
# in bash terminal
ls /Applications/freesurfer/subjects/bert/stats
## aseg.stats
## lh.BA_exvivo.stats
## lh.BA_exvivo.thresh.stats
## lh.aparc.DKTatlas.stats
## lh.aparc.a2009s.stats
## lh.aparc.pial.stats
## lh.aparc.stats
## lh.curv.stats
## lh.w-g.pct.stats
## rh.BA_exvivo.stats
## rh.BA_exvivo.thresh.stats
## rh.aparc.DKTatlas.stats
## rh.aparc.a2009s.stats
## rh.aparc.pial.stats
## rh.aparc.stats
## rh.curv.stats
## rh.w-g.pct.stats
## wmparc.stats
```

```
atlas_data <- read_atlas_files(
  subjects_dir = "/Applications/freesurfer/subjects/",
  atlas = "aparc.stats$"
)
atlas_data
## # A tibble: 204 x 11
##    subject label NumVert SurfArea GrayVol ThickAvg
##    <chr>   <chr>   <int>    <int>   <int>    <dbl>
##  1 bert    lh_b~    1181      831    2297     2.77
##  2 bert    lh_c~     843      572    1534     2.72
##  3 bert    lh_c~    2758     1840    5772     2.80
##  4 bert    lh_c~    2683     1654    3074     1.81
##  5 bert    lh_e~     581      416    1840     3.33
##  6 bert    lh_f~    4113     2875    8519     2.72
##  7 bert    lh_i~    4948     3466   10559     2.70
##  8 bert    lh_i~    5056     3542   12358     2.98
##  9 bert    lh_i~    1561      990    2350     2.09
## 10 bert    lh_l~    7961     5077   12743     2.30
## # ... with 194 more rows, and 5 more variables:
## #   ThickStd <dbl>, MeanCurv <dbl>,
## #   GausCurv <dbl>, FoldInd <int>, CurvInd <dbl>
```

This long format data can be used directly with the `ggseg()`-function, as the 'label' column corresponds in name and content with the 'label' column in the atlas data of `dkt`. The data **must** include a column that has the same column name and at least *some* data matching the values in the corresponding column in the atlas data. In the following example, the `ggseg()`-function will recognise the matching column 'label', and merge the supplied data into the atlas using *dplyr*-joins. The appearance of the plot can then be modified similarly to any other *ggplot2* graph using functions such as scales, labs, themes, etc., as seen in Figure 4

```
# Figure 4A
p1 <- ggseg(.data=someData, mapping=aes(fill=volume))  +
  labs(title="from read_freesurfer_table",
       fill="volume\n(mm^3)") +
  scale_fill_gradient(low="firebrick",high="goldenrod") +
  guides(fill = guide_colourbar(barheight = 3))

# Figure 4B
p2 <- ggseg(.data=atlas_data, mapping=aes(fill=ThickAvg))  +
  labs(title="from read_atlas_files",
       fill="Thickness\n(mean)") +
  scale_fill_gradient(low="forestgreen",high="goldenrod")+
  guides(fill = guide_colourbar(barheight = 3))

(p1 / p2) +
  plot_annotation(tag_levels = "A")
```

If the results are only in one hemisphere, but you still want to plot both of them, make sure your data.frame includes the column 'hemi' with either 'right' or 'left'. In this case, data will be merged into the atlas both by 'area' and by 'hemi'. For more information about adapting data and viewing only one hemisphere or side, the package vignettes contains more elaborate information.

2.1.2 Creating subplots

There is often the need to plot a statistic of interest in different groups (e.g. thickness or brain - cognition relationships in young or older adults). This may be obtained also with `ggseg()`, using *ggplot2*'s

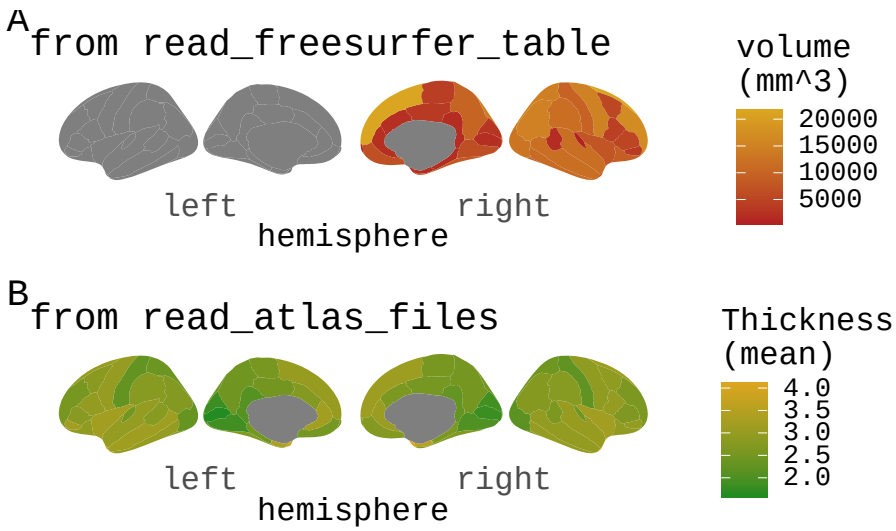

Figure 4. Supplying data through the '.data' option in `ggseg()` enables use of columns in the supplied data to aesthetical arguments (such as 'fill'). The *ggseg* plot can be used with any other polygon compatible function from *ggplot2* or *ggplot2* extentions, for instance adding title, changing the legend name and the colour scheme with standard *ggplot2* functions.

`facet_wrap` or `facet_grid`, with three guiding rules: **1)** as before, data needs to be in long format with a column indexing which group the row corresponds to (group data should appear in seperate rows, not in separate columns). **2)** The data needs to be grouped using *dplyr*'s `group_by()` function *before* providing the data to the `ggseg()`-function. The `ggseg()`-function will detect grouped data, and adapt it to `facet`'s requirements. **3)** Apply `facet_wrap` or `facet_grid` to the plot having used the above two rules. An example of this can be seen in Figure 5 B, where a mock data set including summary statistics for two groups ('Young' and 'Old') is used when faceting a *ggseg*-plot.

In regards to the second rule of needing to group the data before plotting, this is due to the underlying structure of how the provided data and the atlas data are combined before plotting. Since the package function, so far, is a wrapper for {ggplot2} rather than a `geom`, there is certain ggplot2-functionality that is difficult to integrate, like the facets. For the complete atlas-data to be merged with *every data group* correctly, the ggseg function needs to have an indication before plotting what the groups are. The result of ungrouped data into `ggseg` is shown in 5 A. Notice how there are three panels rather than two, and that the regions *not represented in the data* have their own panel with all the grey `NA` colour, while the panels with the data only have the regions represented in the data.

All the concepts described above also work with the `aseg` atlas for subcortical structures, except for 'hemisphere' and 'view' arguments that are superfluous in subcortical atlases (Figure 6).

```
# Make some mock data
someData = tibble(
  area = rep(c("transverse temporal", "insula",
               "pre central","superior parietal"),2),
  p = sample(seq(0,.5,.001), 8),
  AgeG = c(rep("Young",4), rep("Old",4))
)

# Figure 5a
grouped_plot <- someData %>%
  ggseg(position = "stacked",
      mapping=aes(fill=p), show.legend = FALSE) +
  facet_wrap(~AgeG, ncol=3) +
  labs(title = "Ungrouped data")
```

```r
# Figure 5b
ungrouped_plot <- someData %>%
  group_by(AgeG) %>%
  ggseg(position = "stacked",
      mapping=aes(fill=p), show.legend = FALSE) +
  facet_wrap(~AgeG, ncol=3) +
  labs(title = "Grouped data")

# Combining plots with patchwork, applying theme to all
(grouped_plot / ungrouped_plot) +
  plot_layout(widths = 1) +
  plot_annotation(tag_levels = 'A') &
  theme(axis.text = element_text(size=8))
```

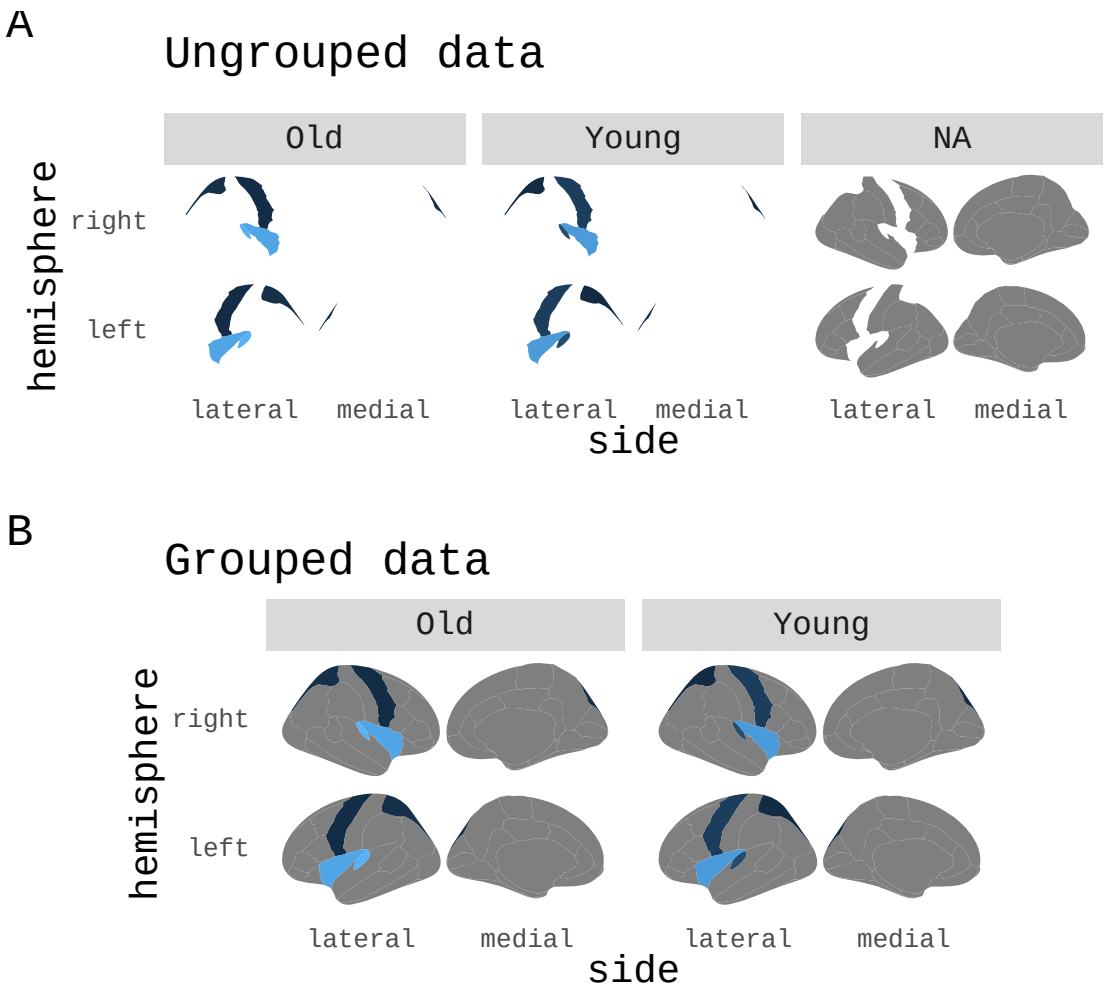

Figure 5. Examples of incorrect and correct faceting with `ggseg`. **A**: ungrouped data will facet incorrectly due to the data not being merged correctly with the atlas. **B**: grouped data will merge in the expected way and result in correctly facetted plots.

The `ggseg` package also has extensive online documentation, with more functionality than we have covered here. The online documentation is also regularly updated with explanations of new features or code improvements.

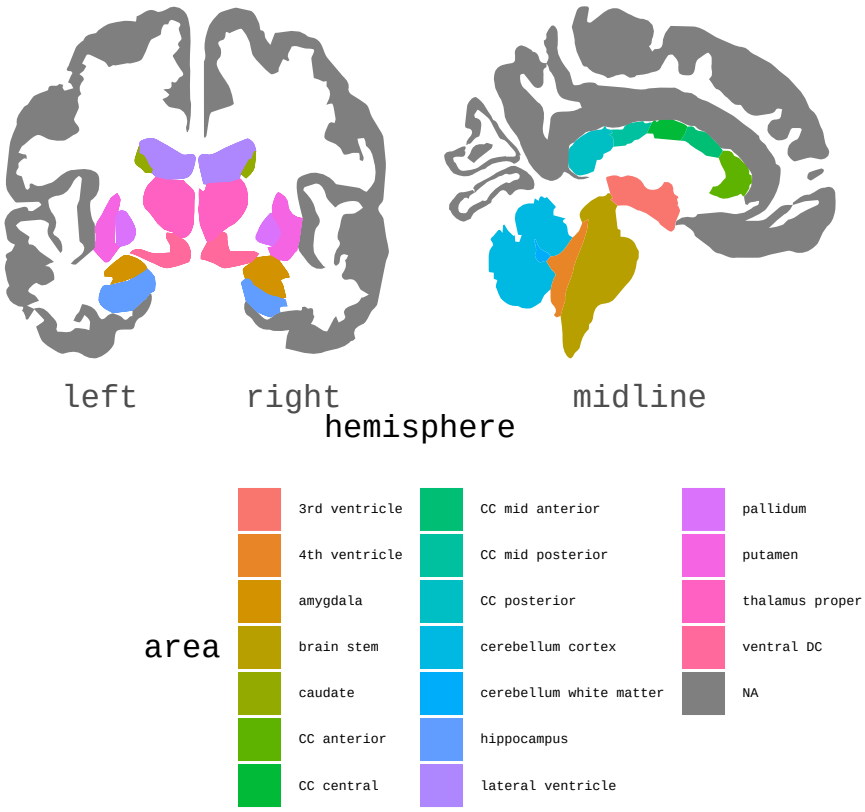

Figure 6. The `aseg` atlas showing subcortical structures has some distinct differences from the `dkt`. For instance, there is no option to show only a single hemisphere. Furthermore, rather than showing lateral and medial surfaces, it shows an axial and sagittal slice.

```
# Figure 6
ggseg(atlas="aseg", mapping=aes(fill=area)) +
  theme(legend.justification=c(1,0),
        legend.position="bottom",
        legend.text = element_text(size = 5)) +
  guides(fill = guide_legend(ncol = 3))
```

2.2 Plotting 3D mesh data

Representing brains as 2D polygons is a good solution for fast, efficient, and flexible plotting, and can be easily combined with interactive apps such as Shiny (Chang et al. [2019]). Yet, brains are intrinsically 3-dimensional and it can be challenging to recognize the location of a region as a flattened image. This problem is exacerbated in atlases that represent subcortical features as they are 3-dimensional, while cortical structures, such as grey matter structures, can be flattened to 2-dimensions. Hence, here we also provide the *ggseg3d* package to plot, view, and print 3D-atlases in R. *ggseg3d* is based on tri-surface mesh plots using *plotly* (Sievert [2018]). The data structure is more complex than the *ggplot2* polygons, and includes additional options for brain inflation, glass brains, camera locations, etc. As *ggseg3d* is based on plotly, the resulting brain atlases are interactive, which guides interpretation, and is useful for public dissemination. We recommend users to familiarize themselves with *plotly* (Sievert [2018]) when using this function.

Out-of-the-box, `ggseg3d()` plots the `dkt_3d` atlas in 'LCBC' surface, but there are two more surfaces available for cortical atlases (Figure 7) The 'LCBC' surface consists on a semi-inflated white matter surface based on the *fsaverage5* template subject. All `[...]_3d` atlases include a `colour` column that based on the color scheme used in the source neuroimaging software.

```r
# Figure 7 left
ggseg3d(surface = "white")  %>%
  pan_camera("right lateral") %>%
  remove_axes() %>%
  layout(title="\nggseg3d(surface='white')",
         font =  list(size = 30))

# Figure 7 center
ggseg3d()  %>%
  pan_camera("right lateral") %>%
  remove_axes() %>%
  layout(title="\nggseg3d(surface='LCBC')",
         font =  list(size = 30))

# Figure 7 right
ggseg3d(surface = "inflated") %>%
  pan_camera("right lateral") %>%
  remove_axes() %>%
  layout(title="\nggseg3d(surface='inflated')",
         font =  list(size = 30))
```

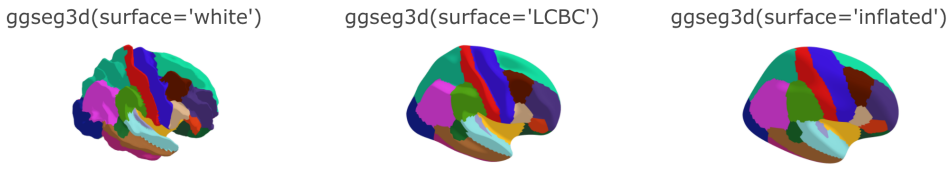

Figure 7. The three surface options provided in *ggseg3d* atlases. **From left to right:** the 'white' surface is the white matter surface, 'LCBC' surface is the semi-inflated white matter surface (inflated over 10 iterations), and the 'inflated' surface is a inflated grey matter cortical surface as provided by the FreeSurfer software. The title for each plot is the literal code use to make the subplot.

The 3D-atlas data is stored in nested tibbles. Each cortical atlas has data sets for three different surfaces (see Figure 7) and the two hemispheres. Only one surface is available for subcortical atlases as inflation procedures are irrelevant. The 'ggseg_3d' column includes all necessary information for `ggseg3d()` to create a 3D mesh-plot, and should not be modified by the user. The additional 3D-atlases in *ggsegExtra* have the same data structure. It is important to note that the coordinates in the plot (X, Y, Z) are **not** any type of radiological coordinate system, but arbitrary Cartesian plot coordinates. While these coordinates are important when running imaging analyses and for reporting of the placement of these regions compared to other regions, this information is not preserved in `ggseg3d`, at the moment. As such, information regarding referencing the regions' location in standard radiological spaces (like MNI or Taliarch) should be acquired directly from imaging software or papers describing the atlases, not `ggseg3d`.

```r
# remotes::install_github("LCBC-UiO/ggseg3d")
library(ggseg3d)
```

2.2.1 External data supply

Similarly as in the 2D-atlas, the user will use `ggseg3d()` to display through a colour scale some descriptive or inferential statistic. If the data is not already in the correct long format or uses similar naming as the atlas, the users should inspect the atlas data for a specific surface (and hemisphere, if desired), and then `unnest(ggseg_3d)` it to see how the atlas data is organised.

```r
# Select surface and hemisphere, and then unnest to inspect the atlas data
dkt_3d %>%
  filter(surf == "inflated" & hemi == "right") %>%
  unnest(ggseg_3d) %>%
  head(5)
```

Note the `mesh` column, which contains lists. Each list corresponds to a region and contains 6 vectors required to create the mesh of the tri-surface plot. It should also be noted that the 'label', 'annot' and 'area' columns could provide matching values for your own data. Similarly to the `ggseg()`-function, the 'label' column should match the region names used in the original neuroimaging software while 'area' and 'annot' provide alternative/secondary names. It is thus important to match your regional identifiers with those used in the atlas. To colour the segments using a column from the data, a column name from the data needs to be supplied to the `colour` option, and providing it to the `text` option will add another line to the *plotly* hover information.

```r
# Figure 8 left
ggseg3d(.data = someData, atlas = dkt_3d, colour = "p", text = "p")

# Figure 8 middle
ggseg3d(.data = someData,
        atlas = dkt_3d,
        colour = "p", text = "p",
        palette = c("#ff0000", "#00ff00", "#0000ff")) %>%
  pan_camera("right lateral") %>%
  remove_axes()

# Figure 8 right
ggseg3d(.data = someData,
        atlas = dkt_3d,
        colour = "p", text = "p",
        na.colour = "black") %>%
  pan_camera("right lateral") %>%
  remove_axes()
```

Figure 8. Left: Supplying data to `ggseg3d()` works similarly to `ggseg()`. Since `ggseg3d()` is based on *plotly* rather than *ggplot2*, most aesthetic adaptations must be set in the main function. Here we set the parcellation colours with the 'colour' column and accompanying mouse-hover text with the 'text' column. **Middle:** The palette for *ggseg3d* needs to be specified directly in the main plot call. The 'palette' option takes vectors of colours either as HEX-codes or R-colour names. **Right:** The colour of the `NA` values can also be changed through the option 'na.colour'.

2.2.2 Customizing colours and the colour bar

You can provide custom colour palettes either in hex or R-names, as seen in Figure 8. Colours will be evenly spaced when creating the colour-scale. A palette may also be supplied as a *named numeric vector*, where the vector names are the colours that users wish to use, and the numeric values are the breakpoints for each colour (e.g. `c("red" = 0, "white" = 0.5, "blue" = 1)`). This way the users can control the minimum and maximum values of the colour scale, and also how the gradient is applied. If

another colour than the default gray is wanted for the NA regions, supply 'na.colour', either as HEX colour or colour name. This option only takes a single colour.

2.2.3 Adding a glass brain

A glass brain is a translucent representation of the brain that can provide a frame of reference, particularly when looking at subcortical parcellations. Since individual brain areas or subcortical structures can be very hard to place without the visual reference of the entire brain, glass brains are often used to aid the viewer understand a regions location. Subcortical atlases include cortical surfaces and other landmark structures for visualization purposes only. One can control the opacity of the these NA structures, to improve visualization, directly in the ggseg3d call. A cortical glass brain can be added to ggseg3d plots through the function add_glassbrain() (Figure @ref(fig:glass brain1)), which takes three extra arguments: hemisphere, colour, and opacity. We recommend only using the glass brain function when plotting subcortical atlases. An interactive version of this plot can be found at https://lcbc-uio.github.io/ggseg3d/articles/ggseg3d.html#colours

```
# Figure 9
ggseg3d(atlas = aseg_3d,
        na.alpha= .5) %>%
  add_glassbrain("left") %>%
  pan_camera("left lateral") %>%
  remove_axes()
```

(ref:glass brain1-cap) For subcortical structure visualization, one can add a glass brain to the plot. This will help with locating the structures relative to the cortex, and make the plot easier to interpret. The glass brain is controlled by three options: opacity, hemisphere, and colour.

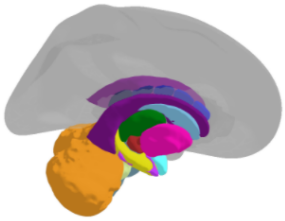

Figure 9. (ref:glassbrain1-cap)

ggsed3d() is based on *plotly* and thus additional *plotly* functionalities can be used to modify and improve the 3D atlas representations. In addition to Carson Sievert's book on *plotly* in R ([2018]), we recommend resources for modifying axes in 3D plots (Damiba [2019a]), the basic introduction to tri-surface plots (Damiba [2019b]), and this tutorial on tri-surface plots with *plotly* in R (Riddihiman [2016]). Finally, we recommend orca command line tool to save *ggseg3d* atlas snapshots.

The package online documentation also includes interactive versions of many of the figures in this tutorial, and several more options not covered here. The online documentation will also be updated as new features are required or we improve the current state of user functionality and options.

2.3 Additional atlases

The *ggseg* and *ggseg3d* packages have two atlases each, which are 2D and 3D variations of the same main atlases: the dkt (Desikan et al. [2006]) and aseg (Fischl et al. [2002]) atlases. These are, however, only

two among many meaningful ways of segmenting the brain into different regions. Thus, the *ggsegExtra* package is a repository containing additional with additional data sets for plotting with the *ggseg* and *ggseg3d* packages. There is an ever-increasing amount of new atlases being created, as research and methods in neuroimaging analysis progresses. The *ggsegExtra*-package is intended to be expanded as a community-effort, as new and informative atlases are published. A small collection of the atlases currently in the *ggsegExtra* package may be viewed in Figure 10, and explored in our shinyapps.io demonstration.

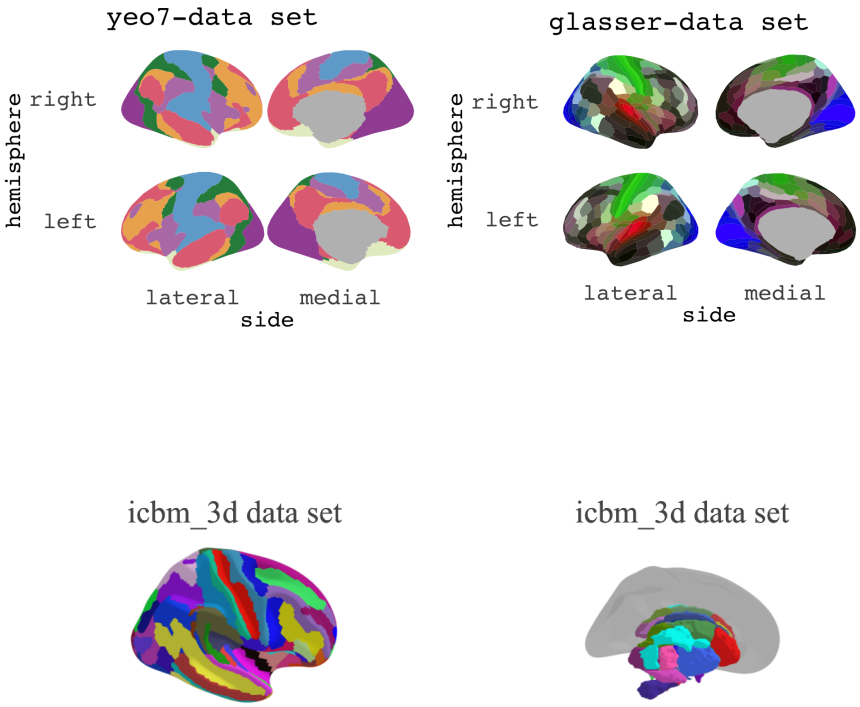

Figure 10. Four example data sets from the *ggsegExtra* library, plotted with the `ggseg()` and `ggseg3d()` functions.

3 DISCUSSION

The main aim of the *ggseg*, *ggseg3d*, and *ggsegExtra* packages is to ease and streamline visualization of brain atlas data in R, by gathering a collection of atlases from several scientific sources and providing customized plotting functions. In this tutorial, we introduced the packages to the readers by presenting some use examples and highlighting the main functions and options that are available. As visualization tools, these packages add up to manifold functionalities such as ggBrain (Fisher [2019]) and ggneuro (Muschelli [2017]) in R, and software-specific image viewers such as FSLeyes (McCarthy [2019]) and Freeview (Dale et al. [1999]). In this regard, we do not aim to compete with software-specific visualizations or advocate for the superiority of the *ggseg*-packages as visualization tools. After all, flattened 2D polygons do not rely on a meaningful brain coordinate system and the units of information in 3D meshes are limited to the number of parcellations. On the contrary, we believe the *ggseg* niche among visualization tools resides in its simplicity and its ability to be combined with statistical analysis pipelines. The possibility to serve as an interactive tool for dissemination and reproducibility when combined with other technologies, such as Binder (Project Jupyter et al. [2018]) or Shiny (Chang et al. [2019]), is an added benefit. This is exemplified in the online supplementary information of Vidal-Piñeiro et al. ([2019]).

The three *ggseg*-packages contain three main features:

1) a collection of 2D-polygon and 3D mesh brain parcellation atlases. The atlas data include the necessary coordinates for plotting, and include other information that should be recognizable for users. **2)** `ggseg()` and `ggseg3d()` functions for visualization. Both functions are flexible and well-adapted to their environment and can be combined with any additional argument from *ggplot2* and *plotly*, respectively. - `ggseg()` is a wrapper function for `geom_polygon` from *ggplot2* and it can be built upon like any *ggplot2* object. - `ggseg3d()` is a *plotly* wrapper function for tri-surface mesh plots which prints 3D atlases. **3)** Complimentary features – e.g. color scales - and functions such as `as_ggseg_atlas()` and `as_ggseg3d_atlas()` to convert data in the correct atlas format.

These functions provide users with the possibility of adapting the plots to their wishes, and also makes it possible to create and contribute to the atlas repository in *ggsegExtra*.

The foundations of the *ggseg*-packages trace back to the necessity of visualizing and exploring the lifespan trajectories of cortical thickness across different brain regions (see supplementary information in Vidal-Piñeiro et al. ([2019])) . That is, *ggseg* appears with the need to inspect and display brain information over time - i.e. including a spatial dimension and a time-varying factor - overcoming the constrains of printed journals and classical 2D plots (e.g. bar plots). The current state of science requires researches to share the results of studies in both high detail and in an intuitive manner, as it permits communication to wide audiences and facilitates reproducibility. Hence, we believe this tool conforms to the essence of open science and invite users to improve the code, provide examples, or tutorials, and contribute to the atlas collection according to their own interest and needs via the public ggseg GitHub repository, ggseg3d GitHub repository, and ggsegExtra GitHub repository.

Finally - while the *ggseg*-packages are circumscribed to brain parcellations - we believe that the structure and functions of the package can be easily applied to any scientific field that benefits from data being displayed across the spatial dimension. We encourage readers to borrow the package functionalities and adapt it to their respective fields and structures of interest, such as has already been done with the *gganatogram*-package (Maag [2018]).

4 PLANNED PACKAGE IMPROVEMENTS

In the *ggsegExtra* github wiki, we offer a pipeline to create and supply atlases for 2D plotting. At the moment, the creation of atlas for ggseg is convoluted and difficult and requires manual intervention. We are in the process of designing a simple, straightforward pipeline to facilitate the creation of new ggseg (2D) atlases. The aim is to create a set of functions that will call specialized tools like FSL(Woolrich et al. [2009]), Freesurfer(Fischl et al. [1999], Dale et al. [1999], Fischl and Dale [2000]) and Imagemagick(Ooms [2019]) to detect the polygon vectors or the mesh segments given an MRI image containing a parcellation specification, and organize these into valid *ggseg* and *ggseg3d* atlases.
Additionally, while `ggseg` works quite nicely as a `ggplot2` wrapper, ideally it should be a proper `geom` or `stat`, making it a `ggplot2` extension rather than a wrapper. This would benefit the user as it would behave more like other `ggplot2` `geoms` and there would be less new functionality to learn. Furthermore, a `geom` would make the process of having to group data *before* faceting become unnecessary, and as such also ease use.
We also hope to make it possible to colour each individual mesh triangle in `ggseg3d`, and as such provide the users with more flexibility in the plotting. This should also improve the speed of `ggseg3d`, which currently is quite slow due to each parcellation region being loaded into `plotly` sequentially. If the entire brain mesh is loaded once, and then the mesh triangles coloured, this would greatly improve speed. Additionally, the current procedure makes it necessary to store the 3d-rendered mesh information directly in the package for all atlases, which also takes up a lot of disk space and makes it slow to download. When able to colour the mesh triangles directly, rather than each parcellation individually, only text data indexing mesh-locations would be necessary and thus reduce package size considerably. We encourage users to contribute to the *ggsegExtra* brain atlas repository by including additional brain atlases.

5 CONCLUSION

Visualization is a fundamental aspect of neuroimaging to explore and understand data, guide interpretation, and communicate with colleagues and the general audience. In this tutorial, we have introduced the *ggseg*-packages, tools for visualizing brain statistics through brain parcellation atlases in R. This visualization

tool easily combines with interactive routines as well as with diverse statistical analysis pipelines. We hope this tool and tutorial proves useful to neuroscientists and inspires others to apply the functions in a wide variety of fields and structures.

6 AUTHOR CONTRIBUTIONS

Didac Vidal-Piñeiro generated the idea for the tool, and the initial scripts for plot visualization. He has also been responsible for converting images from neuroimaging data to ggseg-like data (e.g. polygons and mesh data). Athanasia M. Mowinckel adapted the initial scripts and made the functions into package format and has continued developing the functions with the aim of increasing user-friendliness. She is also responsible for conceiving and adding the mesh-plot functionality through plotly, and developing the pipeline for making that possible. A. M. Mowinckel wrote the first draft of the paper, and both have since critically edited it.

7 CONFLICTS OF INTEREST

The authors declare that there were no conflicts of interest with respect to the authorship or the publication of this article.

8 ACKNOWLEDGEMENTS

We thank John Muschelli for package code comments and its adaptation to neuroconductor ([2018]), and Richard Beare for the first community created atlas, and base code comments. We are indebted to A. M. Winkler scripts for converting neuroimaging data (Freesurfer) into .ply files, a necessary step for neuroimaging to mesh-plots conversion, and to Inge Amlien for the 'LCBC' surface files and helpful contributions to the ggseg3d pipeline. We are indebted to those users that actively contributed to the package and all the individuals that worked on the framework on which the package is sustained (e.g. R-project, neuroimaging software). Finally, we thank Anders M. Fjell (AMF) and Kristine B. Walhovd (KBW) for their encouragement and financial support.

9 FUNDING

This work is funded by EU Horizon 2020 Grant 'Healthy minds 0-100 years: Optimizing the use of European brain imaging cohorts (Lifebrain)', with grant agreement 732592. The project has also received funding from the European Research Council's Starting (grant agreements 283634, to A.M.F. and 313440 to K.B.W.) and consolidator Grant Scheme (grant agreement 771355 to KBW and 725025 to AMF). The project has received funding through multiple grants from the Norwegian Research Council"

10 PRIOR VERSIONS

Some of the content presented here also appears in the package vignettes of *ggseg* and *ggseg3d*, which may be accessed through R or in the package websites (Mowinckel and Vidal-Piñeiro [2019a], Mowinckel and Vidal-Piñeiro [2019b]). Athanasia Monika Mowinckel also has several tutorials on her blog regarding *ggseg* creation and functionality (Mowinckel [2018a], Mowinckel [2018b]).

REFERENCES

Inge K. Amlien, Markus H. Sneve, Didac Vidal-Piñeiro, Kristine B. Walhovd, and Anders M. Fjell. Elaboration Benefits Source Memory Encoding Through Centrality Change. *Scientific Reports*, 9(1): 3704, March 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-39999-1. URL `https://doi.org/10.1038/s41598-019-39999-1`.

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2019. URL `https://CRAN.R-project.org/package=shiny`. R package version 1.3.1.

Chi-Hua Chen, Mark Fiecas, E. D. Gutiérrez, Matthew S. Panizzon, Lisa T. Eyler, Eero Vuoksimaa, Wesley K. Thompson, Christine Fennema-Notestine, Donald J. Hagler, Terry L. Jernigan, Michael C.

Neale, Carol E. Franz, Michael J. Lyons, Bruce Fischl, Ming T. Tsuang, Anders M. Dale, and William S. Kremen. Genetic topography of brain morphology. *Proceedings of the National Academy of Sciences*, 110(42):17089–17094, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1308091110. URL https://www.pnas.org/content/110/42/17089.

Anders Dale, Bruce Fischl, and Martin I. Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9(2):179 – 194, 1999.

Joseph Damiba. plotly axes. https://plot.ly/r/axes/#modifying-axes-for-3D-plots, 2019a. Accessed: 2019-04-01.

Joseph Damiba. plotly tri-surf. https://plot.ly/r/trisurf/, 2019b. Accessed: 2019-04-01.

Rahul S. Desikan, Florent Ségonne, Bruce Fischl, Brian T. Quinn, Bradford C. Dickerson, Deborah Blacker, Randy L. Buckner, Anders M. Dale, R. Paul Maguire, Bradley T. Hyman, Marilyn S. Albert, and Ronald J. Killiany. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *NeuroImage*, 31(3):968 – 980, 2006. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2006.01.021. URL http://www.sciencedirect.com/science/article/pii/S1053811906000437.

Christophe Destrieux, Bruce Fischl, Anders Dale, and Eric Halgren. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *NeuroImage*, 53(1):1 – 15, 2010. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2010.06.010. URL http://www.sciencedirect.com/science/article/pii/S1053811910008542.

Simon B. Eickhoff, B. T. Thomas Yeo, and Sarah Genon. Imaging-based parcellations of the human brain. *Nature Reviews Neuroscience*, 19(11):672–686, November 2018. ISSN 1471-0048. doi: 10.1038/s41583-018-0071-7. URL https://doi.org/10.1038/s41583-018-0071-7.

Bruce Fischl and Anders M. Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences of the United States of America*, 97(20):11050–11055, 2000.

Bruce Fischl, Martin I. Sereno, and Anders Dale. Cortical surface-based analysis: Ii: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*, 9(2):195 – 207, 1999.

Bruce Fischl, David H. Salat, Evelina Busa, Marilyn Albert, Megan Dieterich, Christian Haselgrove, Andre van der Kouwe, Ron Killiany, David Kennedy, Shuna Klaveness, Albert Montillo, Nikos Makris, Bruce Rosen, and Anders M. Dale. Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341 – 355, 2002. ISSN 0896-6273. doi: https://doi.org/10.1016/S0896-6273(02)00569-X. URL http://www.sciencedirect.com/science/article/pii/S089662730200569X.

Aaron Fisher. *ggBrain: ggplot Brain Images*, 2019. R package version 0.1.

Matthew F. Glasser, Timothy S. Coalson, Emma C. Robinson, Carl D. Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F. Beckmann, Mark Jenkinson, Stephen M. Smith, and David C. Van Essen. A multi-modal parcellation of human cerebral cortex. *Nature*, 536:171 EP –, 07 2016. URL https://doi.org/10.1038/nature18933.

Kurt Hornik. The comprehensive r archive network. *WIREs Computational Statistics*, 4(4):394–398, 2012. doi: 10.1002/wics.1212. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1212.

Kegang Hua, Jiangyang Zhang, Setsu Wakana, Hangyi Jiang, Xin Li, Daniel S. Reich, Peter A. Calabresi, James J. Pekar, Peter C.M. van Zijl, and Susumu Mori. Tract probability maps in stereotaxic spaces: Analyses of white matter anatomy and tract-specific quantification. *NeuroImage*, 39(1):336 – 347, 2008. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2007.07.053. URL http://www.sciencedirect.com/science/article/pii/S105381190700688X.

Jesper Maag. gganatogram: An r package for modular visualisation of anatograms and tissues based on ggplot2. *f1000research*, 2018. URL https://f1000research.com/articles/7-1576/v1. Version 1: Awaiting peer review.

Nikos Makris, Jill M. Goldstein, David Kennedy, Steven M. Hodge, Verne S. Caviness, Stephen V. Faraone, Ming T. Tsuang, and Larry J. Seidman. Decreased volume of left and total anterior insular lobule in schizophrenia. *Schizophrenia Research*, 83(2):155 – 171, 2006. ISSN 0920-9964. doi: https://doi.org/10.1016/j.schres.2005.11.020. URL http://www.sciencedirect.com/science/article/pii/S0920996405004998.

Paul McCarthy. Fsleyes, April 2019. URL https://doi.org/10.5281/zenodo.2630502.

Susumu Mori, S. Wakana, Peter C M van Zijl, and L.M. Nagae-Poetscher. *MRI Atlas of Human White Matter*. Elsevier Science, 2005. ISBN 9780080456164. URL https://www.elsevier.com/books/mri-atlas-of-human-white-matter/mori/978-0-444-51741-8.

A.M. Mowinckel. Get the brain animated! - Dr.Mowinckels blog. https://drmowinckels.io/blog/get-the-brain-animated/, 2018a. Accessed: 2019-04-01.

A.M. Mowinckel. Introducing the ggseg r-package for brain segmentations - Dr.Mowinckels blog. https://drmowinckels.io/blog/introducing-the-ggseg-r-package-for-brain-segmentations/, 2018b. Accessed: 2019-04-01.

A.M. Mowinckel and D. Vidal-Piñeiro. ggseg package website. https://lcbc-uio.github.io/ggseg/, 2019a. Accessed: 2019-11-26.

A.M. Mowinckel and D. Vidal-Piñeiro. ggseg3d package website. https://lcbc-uio.github.io/ggseg3d/, 2019b. Accessed: 2019-11-26.

J. Muschelli, A. Gherman, J. P. Fortin, B. Avants, B. Whitcher, J. D. Clayden, B. S. Caffo, and C. M. Crainiceanu. Neuroconductor: an R platform for medical imaging analysis. *Biostatistics*, Jan 2018.

John Muschelli. *ggneuro: Plotting Functions for Neuroimaging Data in 'ggplot2'*, 2017. R package version 0.5.0.

Jeroen Ooms. *magick: Advanced Graphics and Image-Processing in R*, 2019. URL https://CRAN.R-project.org/package=magick. R package version 2.2.

Diego A. Pizzagalli, Avram J. Holmes, Daniel G. Dillon, Elena L. Goetz, Jeffrey L. Birk, Ryan Bogdan, Darin D. Dougherty, Dan V. Iosifescu, Scott L. Rauch, and Maurizio Fava. Reduced caudate and nucleus accumbens response to rewards in unmedicated individuals with major depressive disorder. *American Journal of Psychiatry*, 166(6):702–710, 2009. doi: 10.1176/appi.ajp.2008.08081201. URL https://doi.org/10.1176/appi.ajp.2008.08081201. PMID: 19411368.

Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, and Carol Willing. Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. In Fatih Akici, David Lippa, Dillon Niederhut, and M Pacer, editors, *Proceedings of the 17th Python in Science Conference*, pages 113 – 120, 2018. doi: 10.25080/Majora-4af1f417-011.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Riddihiman. plotly trisurf2. https://moderndata.plot.ly/trisurf-plots-in-r-using-plotly/, 2016. Accessed: 2019-04-01.

Alexander Schaefer, Ru Kong, Evan M Gordon, Timothy O Laumann, Xi-Nian Zuo, Avram J Holmes, Simon B Eickhoff, and B T Thomas Yeo. Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI. *Cerebral Cortex*, 28(9):3095–3114, 07 2017. ISSN 1047-3211. doi: 10.1093/cercor/bhx179. URL https://doi.org/10.1093/cercor/bhx179.

Carson Sievert. *plotly for R*, 2018. URL https://plotly-r.com.

D. Vidal-Piñeiro, N. Parker, J. Shin, L. French, AP. Jackowski, AM. Mowinckel, Y. Patel, Z. Pausova, G. Salum, Ø. Sørensen, KB Walhovd, T. Paus, and AM Fjell. Cellular correlates of cortical thinning throughout the lifespan. *bioRxiv*, page 585786, January 2019. doi: 10.1101/585786. URL http://biorxiv.org/content/early/2019/03/23/585786.abstract.

Kristine B. Walhovd, Anders M. Fjell, Ivar Reinvang, Arvid Lundervold, Anders M. Dale, Dag E. Eilertsen, Brian T. Quinn, David Salat, Nikos Makris, and Bruce Fischl. Effects of age on volumes of cortex, white matter and subcortical structures. *Neurobiology of Aging*, 26(9):1261 – 1270, 2005. ISSN 0197-4580. doi: https://doi.org/10.1016/j.neurobiolaging.2005.05.020. URL http://www.sciencedirect.com/science/article/pii/S0197458005001673.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Mark W. Woolrich, Saad Jbabdi, Brian Patenaude, Michael Chappell, Salima Makni, Timothy Behrens, Christian Beckmann, Mark Jenkinson, and Stephen M. Smith. Bayesian analysis of neuroimaging data in FSL. *NeuroImage*, 45(1):S173–S186, March 2009. doi: 10.1016/j.neuroimage.2008.10.055. URL https://doi.org/10.1016/j.neuroimage.2008.10.055.

Anastasia Yendiki, Patricia Panneck, Priti Srinivasan, Allison Stevens, Lilla Zöllei, Jean Augustinack, Ruopeng Wang, David Salat, Stefan Ehrlich, Tim Behrens, Saad Jbabdi, Randy Gollub, and Bruce Fischl. Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas of the underlying anatomy. *Frontiers in Neuroinformatics*, 5:23, 2011. ISSN 1662-5196. doi: 10.3389/fninf.2011.00023. URL https://www.frontiersin.org/article/10.3389/fninf.2011.00023.

Thomas Yeo, B. T., Fenna M. Krienen, Jorge Sepulcre, Mert R. Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L. Roffman, Jordan W. Smoller, Lilla Zöllei, Jonathan R. Polimeni, Bruce Fischl, Hesheng Liu, and Randy L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3):1125–1165, 2011. doi: 10.1152/jn.00338.2011. URL https://doi.org/10.1152/jn.00338.2011. PMID: 21653723.