Connectivity

An analysis for fNIRS data with groups

Tyson S. Barrett, PhD

Why something new?

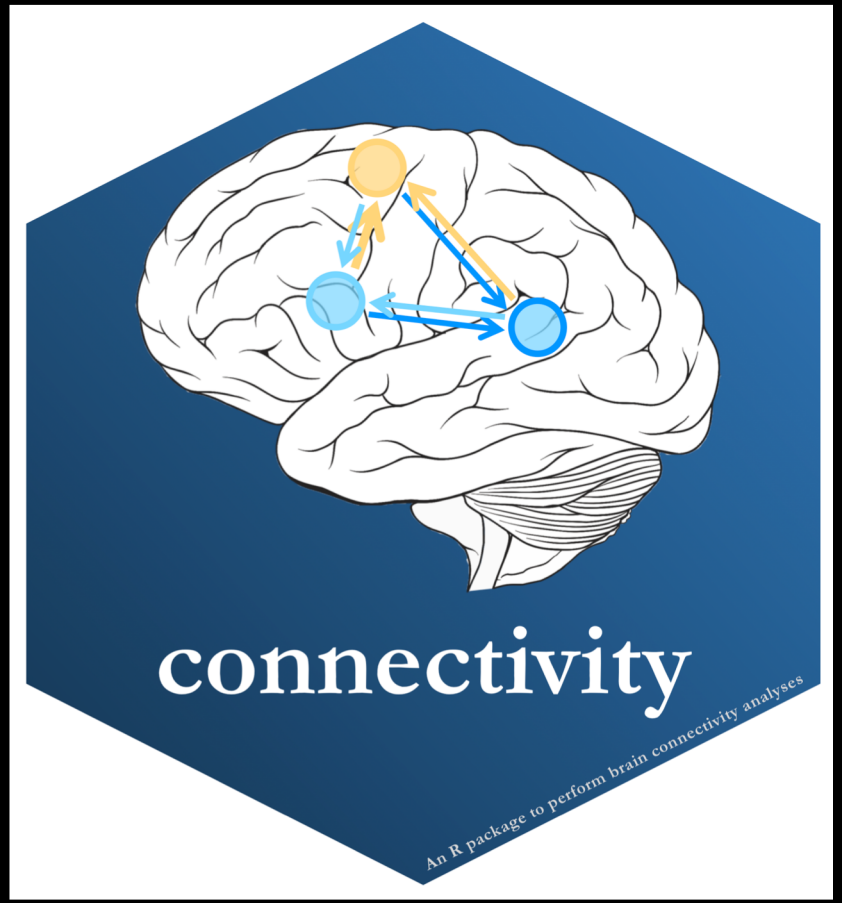
- Effect sizes are important
- Intuitive interpretation is important
- Usability is important
- Flexibility is important
- Reproducibility is important
- Linear mixed effects models are cool

What is connectivity?

- A “recipe” based approach
- Uses linear mixed effect models in a form similar to Granger-Causality models
  - Outcome ~ lag(Outcome) + predictors + (1 | participant)
- Output is in terms of standardized effect sizes that are similar to correlations
- Allows comparisons of groups

How to use?

- Three steps:
  - Import
  - Analyze
  - Visualize



github.com/TysonStanley/connectivity

Import

Currently this is the messiest step and will likely be improved over time

```
-- P07
   |__onset.txt
   |__P07_brodExtract.csv
   |__P07_HBA_Probe1_Oxy.csv
   |__P07_HBA_Probe2_Oxy.csv
-- P08
   |__onset.txt
   |__P08_brodExtract.csv
   |__P08_HBA_Probe1_Oxy.csv
   |__P08_HBA_Probe2_Oxy.csv
-- P09
   |__onset.txt
   |__P09_brodExtract.csv
   |__P09_HBA_Probe1_Oxy.csv
   |__...
```

```r
library(connectivity)

path <- "~/Box/Stuttering Writing Group/PhoneCallsControl/"
data <- import_nirs(path,
                    stg = 22, ipl = c(39, 40), ifg = c(44, 45), sma = 6, m1 = 4)
```

Import

```
library(connectivity)

path <- "~/Box/Stuttering Writing Group/PhoneCallsControl/"
data <- import_nirs(path,
                    stg = 22, ipl = c(39, 40), ifg = c(44, 45), sma = 6, m1 = 4)
```

Gives you data that looks like this

```
data
#> # A tibble: 7 x 2
#>   participant probe_data
#>   <chr>       <list>
#> 1 P01         <tibble [22,239 × 67]>
#> 2 P03         <tibble [23,204 × 67]>
#> 3 P04         <tibble [22,936 × 67]>
#> 4 P05         <tibble [20,056 × 67]>
#> 5 P06         <tibble [30,114 × 67]>
#> 6 P07         <tibble [25,305 × 67]>
#> 7 P16         <tibble [23,486 × 67]>
```

Import

```
library(connectivity)

path <- "~/Box/Stuttering Writing Group/PhoneCallsControl/"
data <- import_nirs(path,
                    stg = 22, ipl = c(39, 40), ifg = c(44, 45), sma = 6, m1 = 4)
```

Gives you data that looks like this

```
data
#> # A tibble: 7 x 2
#>   participant probe_data
#>   <chr>       <list>
#> 1 P01         <tibble [22,239 × 67]>
#> 2 P03         <tibble [23,204 × 67]>
#> 3 P04         <tibble [22,936 × 67]>
#> 4 P05         <tibble [20,056 × 67]>
#> 5 P06         <tibble [30,114 × 67]>
#> 6 P07         <tibble [25,305 × 67]>
#> 7 P16         <tibble [23,486 × 67]>
```

Contains all the data for P01 (including the averages for each region defined above)

Import

To grab just one of the tasks

```
## Subset the data to just resting and assign to `rest`
rest <- data
rest$probe_data = purrr::map(rest$probe_data, ~.x %>% filter(task == "rest"))
## Make sure `rest` still contains information on the regions in the data
attr(rest, "regions")
#> [1] "stg" "ipl" "ifg" "sma" "m1"
```

Import

To grab just one of the tasks

```
## Subset the data to just resting and assign to `rest`
rest <- data
rest$probe_data = purrr::map(rest$probe_data, ~.x %>% filter(task == "rest"))
## Make sure `rest` still contains information on the regions in the data
attr(rest, "regions")
#> [1] "stg" "ipl" "ifg" "sma" "m1"
```

The data should have an attribute called “regions” that has all the regions of interest (this code is how you check)

Analyze

This will run a series of linear mixed effects models, with each region as the outcome predicted by its lag and the regions (plus anything you put in the “covariates”)

“est” is the effect size estimate and “pvalue” is the pvalue (using Satterthwaite approximation to degrees of freedom)

```
fits <- get_connectivity(rest, covariates = c("(1 | participant)"))
fits
#>    outcome rowname           est       pvalue
#> 1      stg     ipl -0.0012854233 4.098717e-02
#> 2      stg     ifg  0.0010528857 2.175877e-01
#> 3      stg     sma  0.0018098305 2.400770e-02
#> 4      stg      m1 -0.0001314380 8.288871e-01
#> 5      stg     lag  0.5951213771 0.000000e+00
#> 6      ipl     stg  0.0068348462 5.309958e-12
#> 7      ipl     ifg -0.0035143519 1.620990e-02
#> 8      ipl     sma  0.0044339404 9.635422e-04
#> 9      ipl      m1  0.0010296516 3.050960e-01
#> 10     ipl     lag  0.5523334334 0.000000e+00
#> 11     ifg     stg -0.0009576724 6.260133e-02
#> 12     ifg     ipl -0.0010242724 5.790339e-02
#> 13     ifg     sma  0.0053455916 7.133882e-14
#> 14     ifg      m1 -0.0030414322 3.975775e-09
#> 15     ifg     lag  0.5315290144 0.000000e+00
#> 16     sma     stg  0.0008315454 4.110240e-02
#> 17     sma     ipl  0.0018432310 1.563944e-05
#> 18     sma     ifg  0.0069252798 6.357022e-27
#> 19     sma      m1  0.0034547031 1.960396e-17
#> 20     sma     lag  0.2008891863 0.000000e+00
#> 21      m1     stg -0.0014515723 9.122225e-03
#> 22      m1     ipl -0.0038264659 6.296958e-11
#> 23      m1     ifg -0.0017079992 4.328194e-02
#> 24      m1     sma  0.0085802104 1.031991e-27
#> 25      m1     lag  0.2522870952 0.000000e+00
```

Visualize

The brain_viz() function will plot the effect sizes (size of arrows) and pvalue (transparent is not significant at .05)

Can control all elements of the visual (the image, where the circles are, etc.)

```
brain_viz(fits)
```