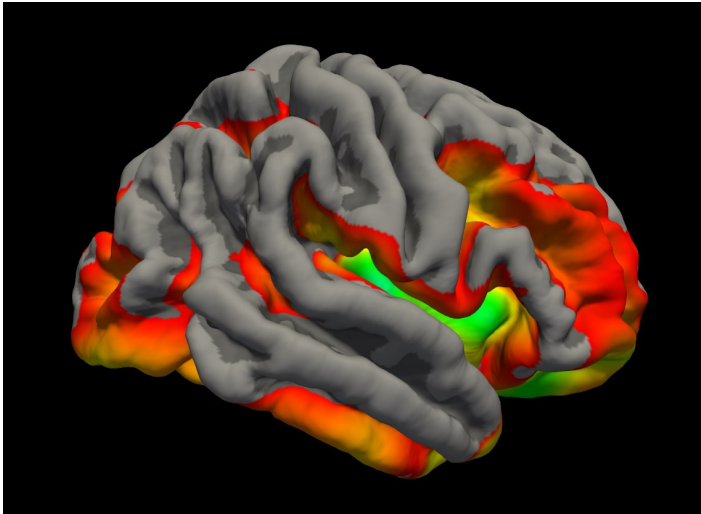


A QUICK GUIDE TO QDECR



Sander Lamballais

Henning Tiemeier

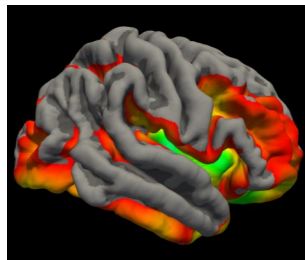
Meike W. Vernooij

M. Arfan Ikram

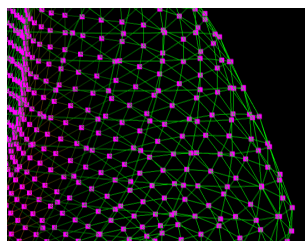
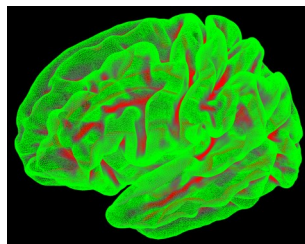
Ryan L. Muetzel

BACKGROUND INFORMATION

The cerebral cortex is integral to brain function and dysfunction. The structure of the cortex is influenced by a myriad of factors, from aging to onset of disease.



Associations between phenotypes and the cerebral cortex can be studied by processing MRI images and capturing the surface of the cortex. This can, for example, be done using the Freesurfer software¹, which reconstructs the cortical surface as thousands of points in 3D space. These points are also known as vertices (singular: vertex). By running an analysis at every vertex, it is possible to map the association of a phenotype with specific brain regions.



Although software for vertex-wise analyses has existed for a long time, they are limited in their statistical features. We solved this by developing QDECOR, a vertex-wise analysis package written in R². R is a programming language with a strong focus on statistical analyses. We designed QDECOR to fully utilize many of R's features: easy model specifications, handling missing and imputed data, and straightforward implementation of new statistical models.

GETTING STARTED WITH QDECR

For the full tutorial, please go to www.qdecr.com!

Before getting started, make you sure have installed:

1. Freesurfer ($\geq 6.0.0$)
2. R ($\geq 3.5.1$)

In R, download and install the devtools package³:

```
install.packages("devtools")
```

Then run:

```
devtools::install_github("slamballais/QDECR")  
library(QDECR)
```

All dependencies will automatically be installed as well.

Running QDECR

If we want to see how sex and age influence cortical thickness for the left hemisphere, we have to run:

```
vw <- qdecr_fastlm(qdecr_thickness ~ sex + age,  
  data = pheno,          # your R phenotype dataset  
  id = "id",             # the name of the id column  
  hemi = "lh",           # left hemisphere; "rh" for right  
  project = "test")      # a name for your project
```

That is it! There are other arguments that you can specify. Type `?qdecr_fastlm` to see them all.

UNDERSTANDING THE OUTPUT

The output is now stored in the `vw` variable. We can inspect it:

```
print(vw) # or just `vw`
```

This displays all information: the input, all the paths, the model and some general output.

We can inspect the significant clusters:

```
summary(vw)
```

variable	cluster	n_vertices	mean_area	mean_coefficient	mean_se
(Intercept)	1	147135	0.5245423	-0.57402478	0.03690280
sex2	1	6857	0.5133509	-0.06353143	0.01576786
sex2	2	6806	0.5364039	-0.07483302	0.01898096
sex2	3	5487	0.5332893	-0.06092796	0.01436787
sex2	4	3914	0.5369445	-0.06913389	0.01892905
sex2	5	784	0.5255159	-0.08986683	0.02466390
sex2	6	1091	0.5335917	-0.05217778	0.01438188
sex2	7	704	0.5252148	-0.06384714	0.01490453

To obtain spatial information, we can add:

```
summary(vw, annot = TRUE) # default is aparc.annot
```

		top_region1	
1	superiorfrontal	(8.28%, 100%)	
2	superiorfrontal	(71.71%, 40.37%)	
3	fusiform	(43.89%, 63.36%)	
4	insula	(25.79%, 27.06%)	
5	supramarginal	(75.52%, 34.37%)	
6	pericalcarine	(78.19%, 32.06%)	
7	precentral	(87.63%, 8.90%)	
8	rostralanteriorcingulate	(63.07%, 32.89%)	
		top_region2	top_region3
1	precentral	(7.30%, 100%)	superiorparietal (7.11%, 100%)
2	rostralmiddlefrontal	(14.45%, 13.68%)	frontalpole (3.97%, 100.00%)
3	lateraloccipital	(25.39%, 27.09%)	parahippocampal (13.96%, 51.69%)
4	superiortemporal	(24.42%, 18.43%)	lateralorbitofrontal (17.40%, 22.80%)
5	postcentral	(15.41%, 6.33%)	superiortemporal (4.98%, 2.68%)
6	cuneus	(21.81%, 10.49%)	<NA>
7	parsopercularis	(12.28%, 4.30%)	postcentral (0.09%, 0.01%)
8	medialorbitofrontal	(36.93%, 9.80%)	<NA>

The first percentage is the size of that region within the cluster.

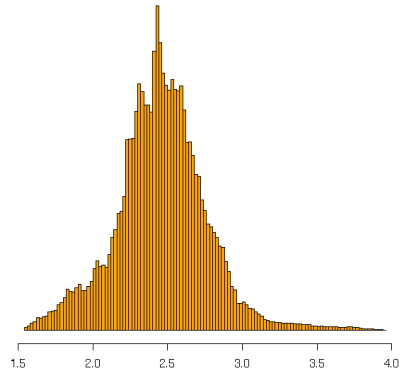
The second percentage is the size of the cluster within the region.

PLOTTING

As of version 0.7.0, there are four options for plotting:

1. `hist(vw, ...)`

Displays a histogram of the mean vertex measure, e.g. cortical thickness, per vertex.

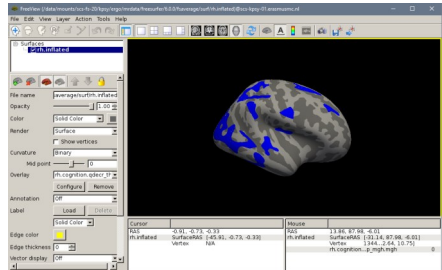


2. `hist(vw, qtype = "subject", ...)`

Displays a histogram of the mean vertex measure, e.g. cortical thickness, per subject.

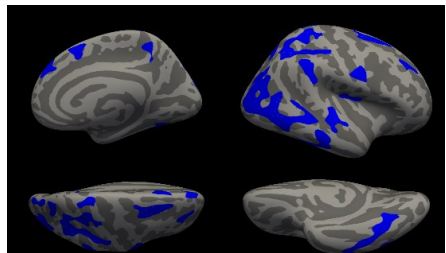
3. `freeview(vw, 2, ...)`

Opens the entered stack/contrast into an interactive Freeview session.



4. `qdecr_snap(vw, 2, ...)`

Entering the stack/contrast number will create snapshots of the thresholded coefficient map using Freeview.



RELEVANT R FEATURES

FORMULAS

R uses formulas to create design matrices. Formula objects are very flexible and allow some interesting options by default:

- Interactions: `qdecr_thickness ~ sex * age`
- Polynomials: `qdecr_thickness ~ poly(age, 2)`
- Splines: `qdecr_thickness ~ splines::ns(age, 3)`
- Arithmetic: `qdecr_thickness ~ I(height / weight^2)`
- Standardization: `qdecr_thickness ~ scale(age)`
- Functions: `qdecr_thickness ~ cut(age, 3)`

IMPUTED DATASETS

Missing data can be imputed with R packages such as `mice`⁴. QDECR automatically recognizes several formats of imputed datasets.

BIG DATA

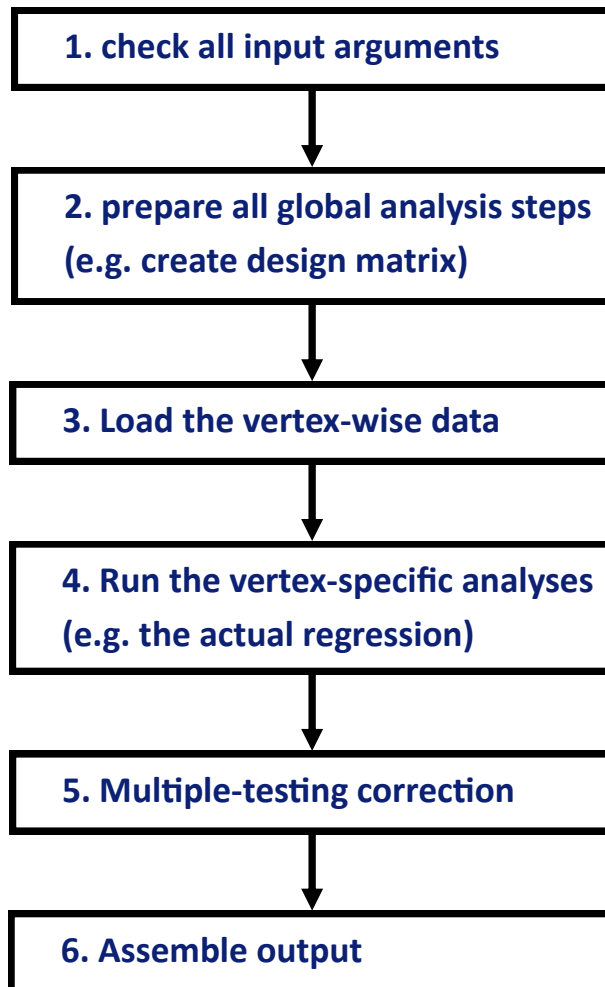
QDECR uses file-backed matrices to reduce the load on the RAM. QDECR was designed to handle datasets containing thousands of participants.

Furthermore, QDECR can be run in parallel across multiple cores/threads. This can be done by specifying the `n_cores` argument:

```
qdecr_fastlm(qdecr_thickness ~ age + sex, n_cores = 4, ...)
```


INTERNALS OF QDECR

QDECR has a modular design. Implementation of new methods (e.g. linear mixed models) simply requires changing steps 2 and 4. All the other things are already taken care of.



CONTRIBUTE

The package is actively being developed. All contributions, ideas and suggestions by anyone are welcome. Instructions on how to contribute to the project can be found on the website.

QDECR is open-source (GPL-3 license).

CONTACT



www.qdecr.com



[slamballais/QDECR](https://github.com/slamballais/QDECR)



lamballais.sander@gmail.com



[@slamballais](https://twitter.com/slamballais)

REFERENCES

1. Fischl B (2012) Freesurfer. *Neuroimage*, 62(2):774-81
2. R Core Team (2018) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org>
3. Wickham H, Hester J & Chang W (2018) devtools: Tools to make developing R packages easier. R package version 2.0.1. <https://CRAN.R-project.org/package=devtools>
4. Van Buuren S & Groothuis-Oudshoorn K (2011). mice: Multivariate Imputation by Chained Equations in R. *J Stat Soft*, 45(3): 1-67