Simulating fMRI data: the R package neuRosim

Marijke Welvaert

Department of Data Analysis
Ghent University

Berlin Workshop on Statistics and Neuroimaging 2011

UNIVERSITEIT GENT

Outline

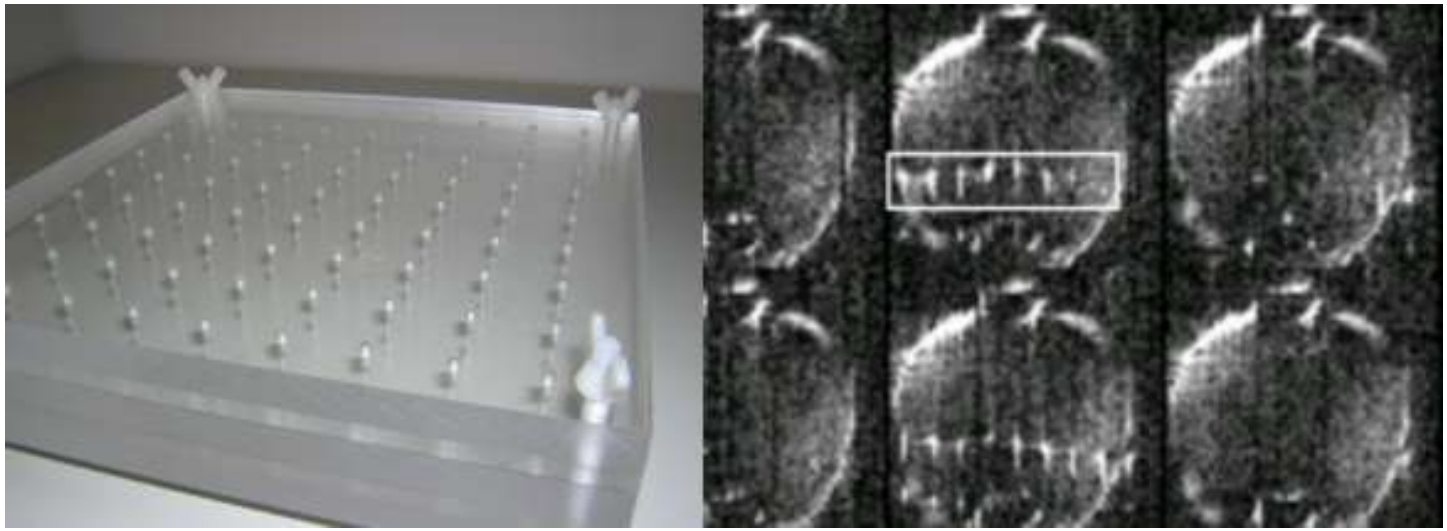
1 Motivation
- The need for simulation
- The need for validated simulation
- The need for software

2 Features
- Goals
- What can you do with neuRosim?
- How is neuRosim organized?

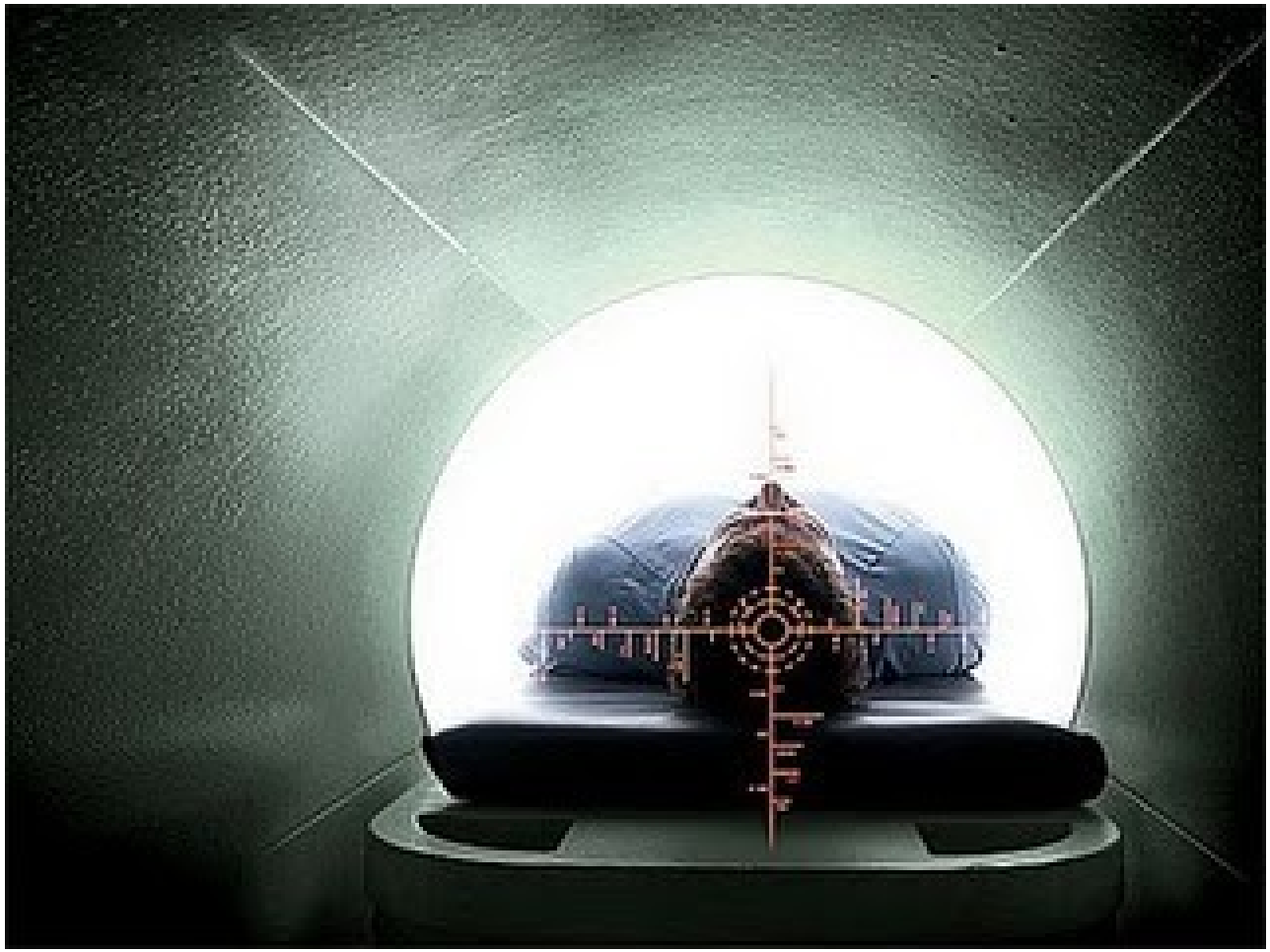
3 Example
- Setting up the design
- Simulating the data
- Exporting and analyzing the data

4 Summary

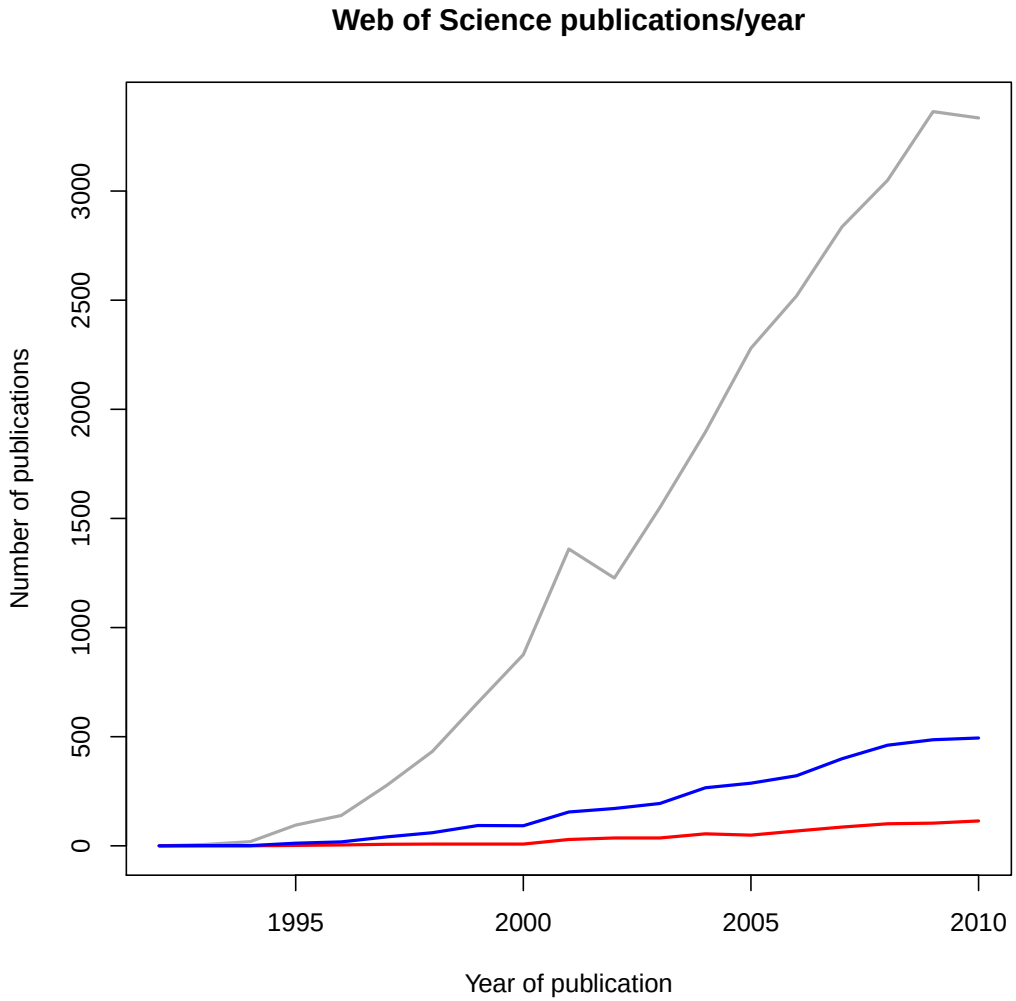
Knowing the ground truth in MRI



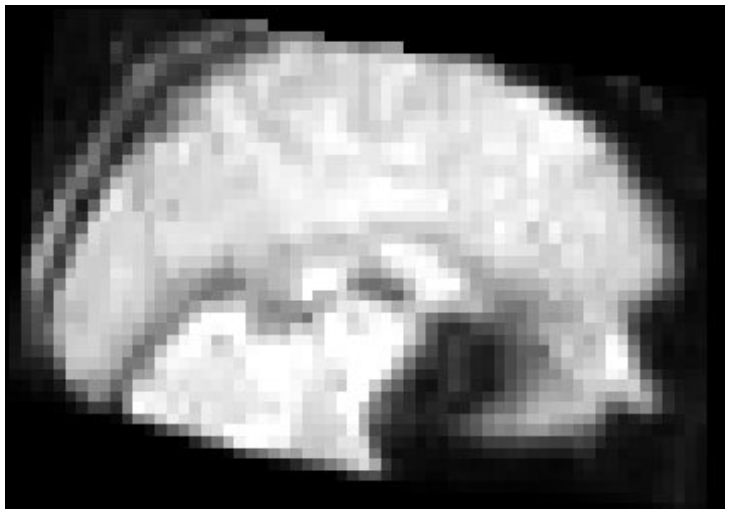
Knowing the ground truth in fMRI



Reflection in the literature



fMRI data components



Activation

- experimentally induced
- spontaneous

Known artefacts

- B_0 inhomogeneities
- low-frequency drift

Noise

- system
- movement
- physiological
- task-related
- ...

Spatial and temporal correlations

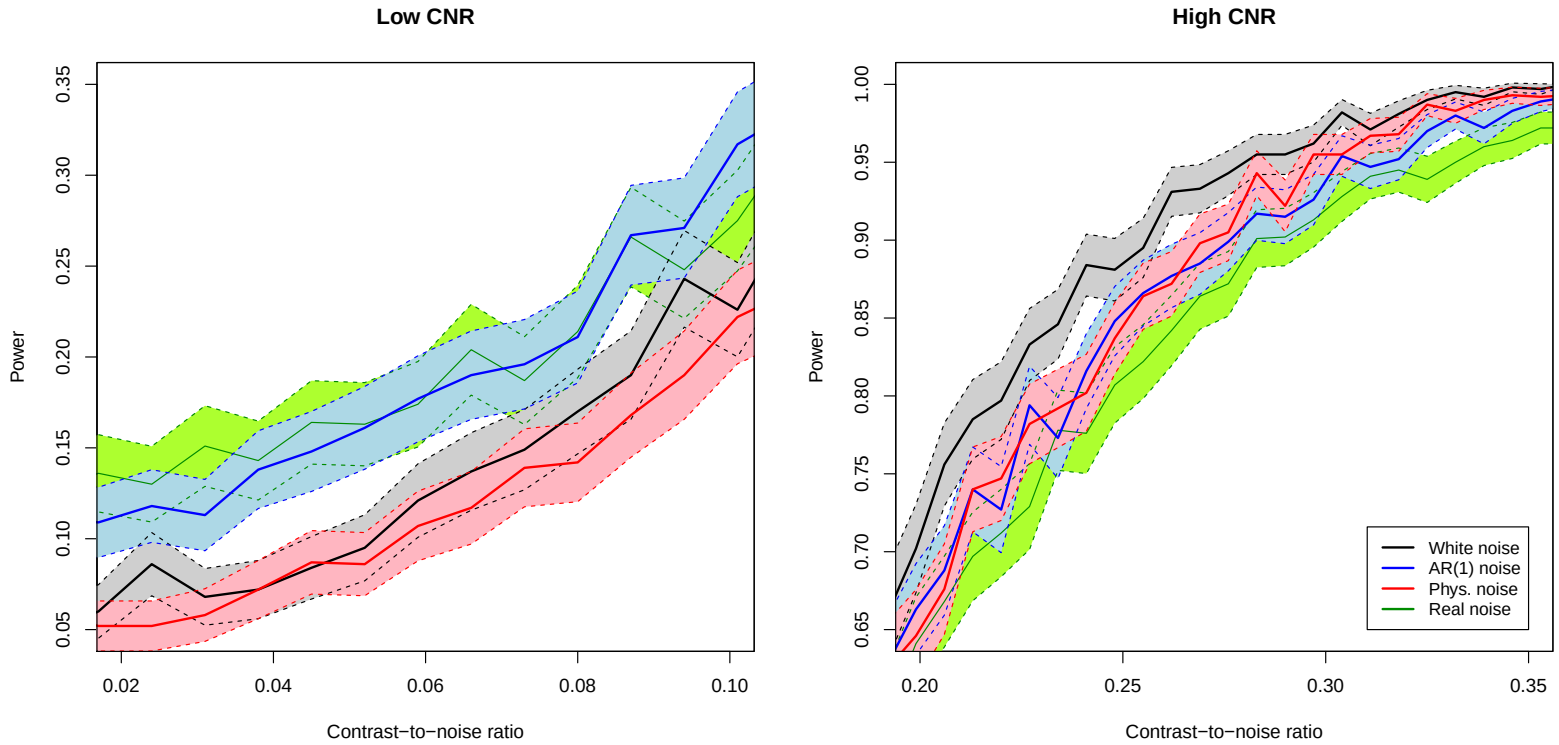
Typical fMRI simulation studies

1. *hybrid simulation*
   - *known* activation combined with *real* noise
   - e.g. Bianciardi *et al.* (2004), Lange (1999), Weibull *et al.* (2008)
2. white time series
   - *known* activation combined with *white* noise
   - i.i.d or AR(1) Gaussian distribution
   - e.g. Lei *et al.* (2010), Lin *et al.* (2010), Purdon & Weisskoff (1998), Smith *et al.* (2011)
3. other
   - model-based simulation, Bloch equations, noise based on residuals of real data
   - e.g. Drobnjak *et al.* (2006), Havlicek *et al.* (2010), Logan & Rowe (2004), Ramsey *et al.* (2010)

Problems - Discrepancies - Shortcomings

- *real* noise may contain undesired activity
- simulated noise = system noise
- beware of the phrase:
  "...simulations under realistic noise conditions..."
- total ignorance of spatial context
- no stand-alone simulations
- often missing (crucial) information while reporting simulation studies

The choice of simulation model matters!



Towards a convergence of simulation methods

- in-house developed software routines, often not available for the community
- language barrier
- no widespread software packages

But. . .

- POSSUM (FSL)
- DCM simulator (SPM)
- simtb (Matlab Toolbox)

UNIVERSITEIT GENT

neuRosim wants

- to provide a tool for simulating fMRI data
- to be a base for more validated simulation studies
- to make simulation available for less technical researchers
- to allow maximum flexibility for the useRs

What can you do with neuRosim?

- specify your experimental design based on stimulus onsets and durations
- specify activated regions using an xyz-coordinate system
- simulate BOLD activation with the choice of different models
- simulate resting state activation *(still under development)*
- simulate fMRI noise originating from different noise sources
- generate fMRI data from 1D time series to 4D volume data

Low-level functions

Building blocks for advanced useRs who want in-depth control over their simulation data

Activation functions

stimfunction()
specifydesign()
specifyregion()

canonicalHRF()
gammaHRF()
balloon()

Noise functions

systemnoise()
temporalnoise()
spatialnoise()
lowfreqdrift()
physnoise()
tasknoise()

High-level functions

Direct simulation of fMRI data

Preparation functions

simprepTemporal()
simprepSpatial()

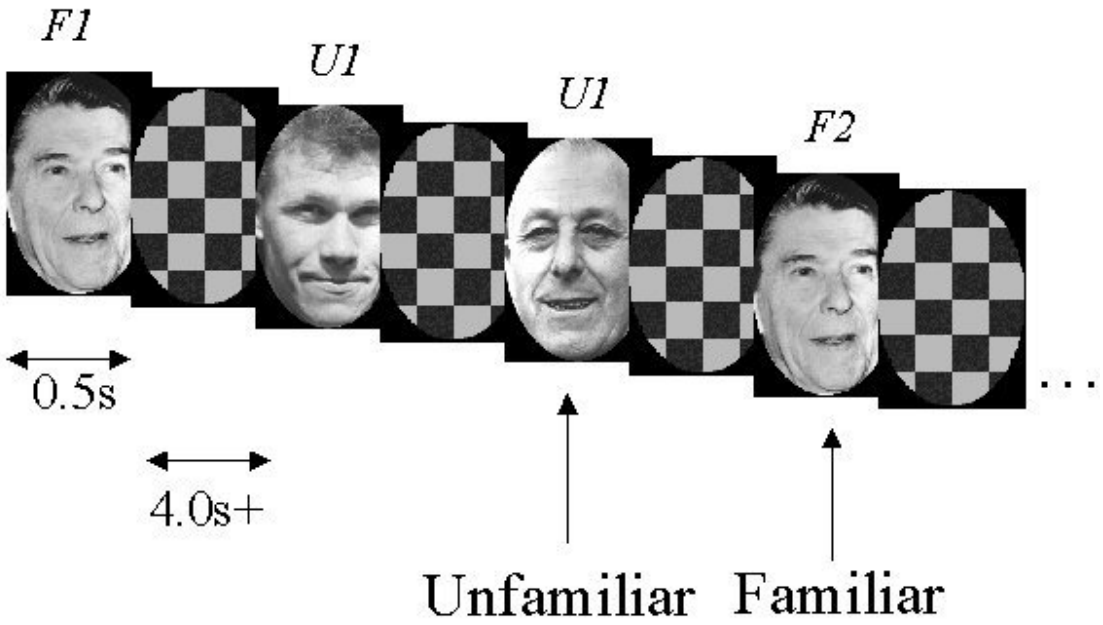
Simulation functions

simTSfmri()
simVOLfmri()
simRestingStatefmri()

Real dataset

Consider the data from a *repetition priming* experiment performed using event-related fMRI (Henson *et al.*, 2002).

- 2×2 factorial design
- famous vs non-famous faces
- effect of repetition



Setting up the design (1)

Temporal Parameters

```
R> nscan <- 351
R> TR <- 2
R> total.time <- nscan*TR
R> onsets.N1 <- c( 6.75,  15.75,  18.00,  27.00,  29.25,  31.50,
+         36.00,  42.75,  65.25,  74.25,  92.25, 112.50, 119.25,
+        123.75, 126.00, 137.25, 141.75, 144.00, 146.25, 155.25,
+        159.75, 162.00, 164.25, 204.75, 238.50)*TR
R> onsets.N2 <- c(13.50,  40.50,  47.25,  56.25,  90.00,  94.50,
+         96.75, 135.00, 148.50, 184.50, 191.25, 202.50, 216.00,
+        234.00, 236.25, 256.50, 261.00, 281.25, 290.25, 303.75,
+        310.50, 319.50, 339.75, 342.00)*TR
R> onsets.F1 <- c( 0.00,   2.25,   9.00,  11.25,  22.50,  45.00,
+         51.75,  60.75,  63.00,  76.50,  78.75,  85.50,  99.00,
+        101.25, 103.50, 117.00, 130.50, 150.75, 171.00, 189.00,
+        227.25, 265.50, 283.50, 285.75, 288.00, 344.25)*TR
R> onsets.F2 <- c(33.75,  49.50, 105.75, 153.00, 157.50, 168.75,
+        177.75, 180.00, 182.25, 198.00, 222.75, 240.75, 254.25,
+        267.75, 270.00, 274.40, 294.75, 299.25, 301.50, 315.00,
+        317.25, 326.25, 333.00, 335.25, 337.50, 346.50)*TR
R> onsets <- list(onsets.N1, onsets.N2, onsets.F1, onsets.F2)
R> dur <- list(0, 0, 0, 0)
```

Setting up the design (2)

Spatial parameters

```
R> region.1A.center <- c(13,13,11)
R> region.1A.radius <- 4
R> region.1B.center <- c(40,18,9)
R> region.1B.radius <- 6
R> region.1C.center <- c(10,45,24)
R> region.1C.radius <- 3
R> region.2.center <- c(15,16,31)
R> region.2.radius <- 5
R> region.3.center <- c(12,16,13)
R> region.3.radius <- 5
R> coord.regions <- list(region.1A.center, region.1B.center,
+       region.1C.center, region.2.center, region.3.center)
R> radius.regions <- c(region.1A.radius,region.1B.radius,
+      region.1C.radius,region.2.radius,region.3.radius)
R> onsets.regions <- list(onsets, onsets, onsets,
+  onsets, onsets)
R> dur.regions <- list(dur, dur, dur, dur, dur)
```

Setting up the design (3)

Conditions per region

- region 1a-b-c: faces versus baseline
- region 2: non-famous versus famous
- region 3: unfamiliar versus familiar (repetition effect)

Effect sizes

```
R> region.1a.d <- list(160.46, 140.19, 200.16, 160.69)
R> region.1b.d <- list(140.51, 120.71, 160.55, 120.44)
R> region.1c.d <- list(120.53, 120.74, 140.02, 100.48)
R> region.2.d  <- list( -0.24,  10.29,  80.18, 160.24)
R> region.3.d  <- list(200.81,  50.04, 240.60,  50.83)
R> effect <- list(region.1a.d,region.1b.d,region.1c.d,
+       region.2.d,region.3.d)
```

Using the high-level functions of neuRosim

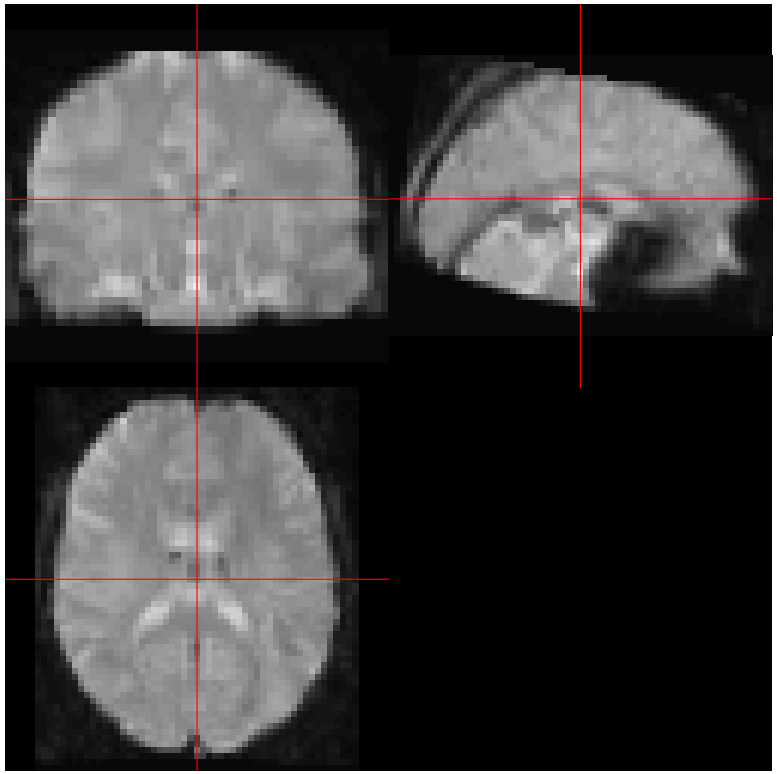
Preparing the spatial and temporal structure

```
R> design <- simprepTemporal(regions=5,
+    onsets=onsets.regions, durations=dur.regions,
+    hrf="double-gamma", TR=TR, totaltime=total.time,
+    effectsize=effect)
R> spatial <- simprepSpatial(regions=5,
+    coord=coord.regions, radius=radius.regions,
+    form="sphere", fading=0.01)
```

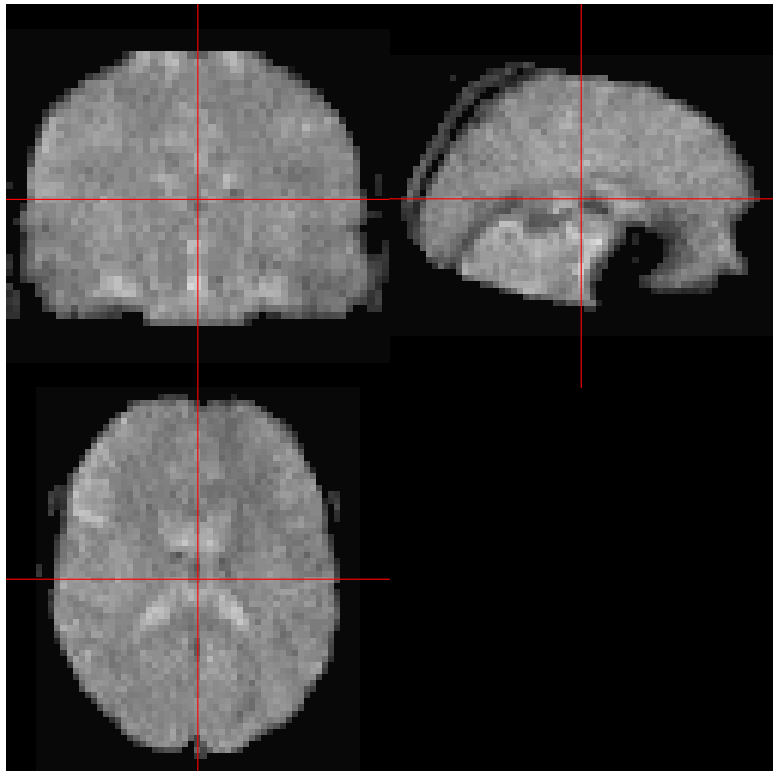
Generating the dataset

```
R> sim.data <- simVOLfmri(design=design, image=spatial,
+    base=baseline, SNR=3.87, noise="mixture", type="rician",
+    rho.temp=c(0.142,0.108,0.084), rho.spat=0.4,
+    w=c(0.05,0.1,0.01,0.09,0.05,0.7), dim=c(53,63,46),
+    template=baseline.bin, spat="gaussRF")
```

Visual comparison of the data



Real data



Simulated data

I have my simulated data, and now. . .

Analysis within R

- Independent Component Analysis: AnalyzeFMRI
- General Linear Model Analysis: fmri

Outside R

Export the data as a NIfTI file with `nifti.image.write` (Rniftilib) or `writeNIfTI` (oro.nifti) and treat as regular dataset

Conclusions

neuRosim provides

- fairly fast simulation of time series to 4D fMRI data
- flexibility for the useRs, both advanced and new
- several activation models
- combination of noise sources

Coming up

- guidelines for validated simulation
- more neurobiological models
  (e.g. Drysdale *et al.*, 2010, Sotero *et al.*, 2009)
- complex-valued fMRI data

Thank you for your attention

Thanks to

Yves Rosseel
Joke Durnez
Beatrijs Moerkerke
Geert Verdoolaeghe

You want to be a useR?

http://cran.r-project.org/web/packages/neuRosim

Marijke.Welvaert@UGent.be

Check out!

Special issue of Journal of Statistical Software: *MRI in R*