Simulating fMRI Data with Realistic Noise using neuRosim

Marijke Welvaert and Yves Rosseel

Department of Data Analysis, Ghent University

Why an fMRI simulation package?

Simulation is often used as method to know the ground truth of the data and is a cheap and fast alternative for invasive procedures like iEEG.

Existing fMRI simulation studies use - almost exclusively - ad-hoc methods and in-house software routines, probably due to the non-existence of validated methods and the lack of widely-available simulators.

What can you do with neuRosim?

* specify your experimental design based on the stimulus onsets
* 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 accounting for different noise sources

* generate fMRI data from time series to 4D data

How is neuRosim organized?

Low-level functions are the building blocks of neuRosim and are intended for advanced users who want in-depth control over their simulated data.

Activation functions

```
stimfunction()
specifydesign()
specifyregion()

canonicalHRF()
gammaHRF()
```

Noise functions

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

High-level functions can simulate fMRI data directly.

Preparation functions

```
simprepTemporal()
simprepSpatial()
```

Simulation functions

```
simTSfmri()
simVOLfmri()
simRestingStatefmri()
```

Example: setting up the design

We consider the data from a repetition priming experiment performed using event-related fMRI (Henson et al. 2002).
* 2x2 factorial design
* famous vs non-famous faces
* effect of repetition

Setting up the temporal parameters (onsets and durations should be in list-format):

```
nscan <- 351
TR <- 2
total.time <- nscan*TR
onsets.N1 <- c(6.75, 15.75, 18, 27, 29.25, 31.5, 36, 42.75, 65.25, 74.25,
    92.25, 112.5, 119.25, 123.75, 126, 137.25, 141.75, 144, 146.25,
    155.25, 159.75, 162, 164.25, 204.75, 238.5)*TR
onsets.N2 <- c(13.5, 40.5, 47.5, 56.25, 90, 94.5, 96.75, 135, 148.5, 184.5,
    191.25, 202.5, 216, 234, 236.25, 256.5, 261, 281.25, 290.25, 303.75,
    310.5, 319.5, 339.75, 342)*TR
onsets.F1 <- c(0, 2.25, 9, 11.25, 22.5, 45, 51.75, 60.75, 63, 76.5, 78.75,
    85.5, 99, 101.25, 103.5, 117, 130.5, 150.75, 171, 189, 227.25,
    265.5, 283.5, 285.75, 288, 344.25)*TR
onsets.F2 <- c(33.75, 49.5, 105.75, 153, 157.5, 168.75, 177.75, 180,
182.25,
    198, 222.75, 240.75, 254.25, 267.75, 270, 274.4, 294.75, 299.25,
301.5,
    315, 317.25, 326.25, 333, 335.25, 337.5, 346.5)
```

Setting up the spatial parameters to define 5 activated regions:

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

Defining the effect size per condition for each region:

```
region.1a.d <- list(160.46, 140.19, 200.16, 160.69)
region.1b.d <- list(140.51, 120.71, 160.55, 120.44)
region.1c.d <- list(120.53, 120.74, 104.02, 100.48)
region.2.d <- list(-0.24, 10.29, 80.18, 160.24)
region.3.d <- list(200.81, 50.04, 240.6, 50.83)
effect <- list(region.1a.d, region.1b.d, region.1c.d, region.2.d,
```

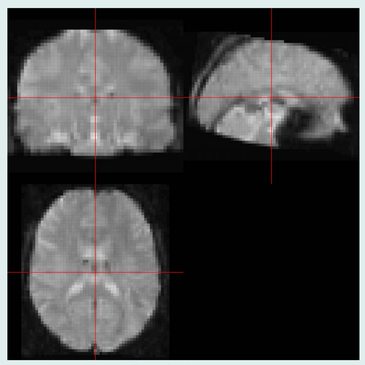
Example: simulating the data

First, we prepare the temporal and spatial structure of our simulated 4D fMRI data using the preparation functions of neuRosim.

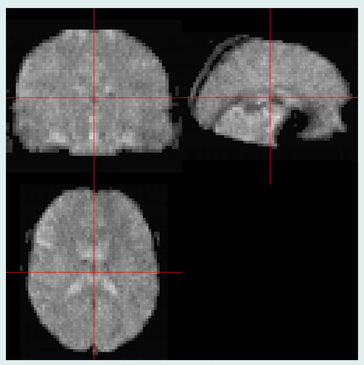
```
design <- simprepTemporal(regions = 5,
  onsets = onsets.regions, durations = dur.regions,
  hrf="double-gamma", TR = TR, totaltime = total.time,
  effectsize = effect)

spatial <- simprepSpatial(regions = 5,
  coord = list(region.1A.center, region.1B.center,
  region.1C.center, region.2.center, region.3.center),
  radius = c(region.1A.radius, region.1B.radius,
  region.1C.radius, region.2.radius, region.3.radius),
  form = "sphere", fading = 0.01)
```

Finally, we can generate the dataset. Note that the values for the SNR and temporal autocorrelation coefficients were estimated based on the real data.

```
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, 43, 46), nscan = 351, vee= 0,
  spat = "gaussRF", template = baseline.bin)
```



Real data



Simulated data

Technicalities

* neuRosim is available for all platforms (Linux, Mac, Windows)
* The Beta version can be downloaded from CRAN (http://cran.r-project.org)
* Including documentation in the form of help pages

References

Henson et al. (2002). Face Repetition Effects in Implicit and Explicit Memory Tests as measured by fMRI. Cerebral Cortex, 12, 178-186

Welvaert et al. (accepted). neuRosim: An R Package for Generating fMRI Data. Journal of Statistical Software.

Contact

E-mail: Marijke.Welvaert@Ugent.be
Website: http://users.ugent.be/~mwelvaer
Address: H. Dunantlaan 1, B-9000 Gent, Belgium