

INTRODUCTION

We present a method based on computer simulations to test the goodness of population receptive field (pRF) estimates. In particular, we have examined the effect of having non-linearity in the hemodynamic response function (HRF) by,

1. Simulating fMRI responses with non-linear "simulation HRFs"
2. Estimating pRFs with a "similar" linear "estimation HRF"

This is an important scenario to test since many estimation methods assume linear HRF models but actual HRF response is non-linear. Hence we present a pipeline to test robustness of stimulation protocols against non-linearities in the BOLD response.

METHOD

We have developed a test based on models in [1][2] with following steps:

1. Parameter Initialization

Define pRF vector $\Theta = (x_0, y_0, \sigma)$. Initialize $\Theta = \theta$ such that x_0, y_0 are receptive field locations in the visual field corresponding to a voxel measurement and the pRF size is modeled accordingly to account for cortical mapping,

$$\sigma = \frac{1}{2} \ln(e + \sqrt{x_0^2 + 2y_0^2})$$

2. Data Generation

$$g(x, y, \Theta = \theta) = e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$$

$$r(t, \Theta = \theta) = \left(\sum_{x,y} s(x, y, t) g(x, y, \theta) \right)^n$$

$$y(t) = h_s(r(t)) + e(t)$$

3. Parameter Estimation

$$r(t, \Theta) = \left(\sum_{x,y} s(x, y, t) g(x, y, \Theta) \right)^n$$

$$p(t) = h_e(r(t, \Theta))$$

$$\hat{\Theta} = \arg \min_t (y(t) - p(t))^2$$

4. Accuracy Map Evaluation

$$\tilde{x} = \frac{|\hat{x}_0 - x_0|}{x_0}, \tilde{y} = \frac{|\hat{y}_0 - y_0|}{y_0}, \tilde{\sigma} = \frac{|\hat{\sigma} - \sigma|}{\sigma}$$

with following variable descriptions:

x, y = point in visual space

n = spatial linearity factor

$g(x, y, \Theta)$ = pRF model

$s(x, y, t)$ = stimulation function

$r(t)$ = pRF response

$e(t)$ = Gaussian noise

$y(t)$ = fMRI response

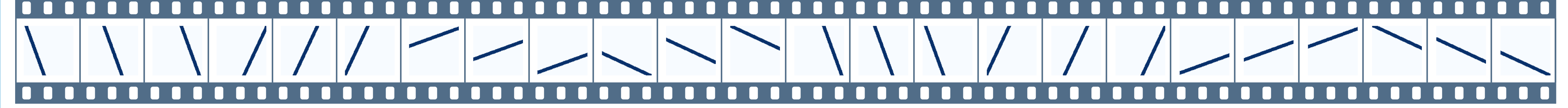
$h(\cdot)$ = HRF function

STIMULATION

We first used the exact drifting bar stimulation in [1]:



Then we optimized it for angle and bar width to reduce the non-linearity:



EXPERIMENT

We use following HRFs:

- Double-Gamma Linear HRF

$$p(t) = r(t) * h(t)$$

$$h(t) = \frac{1}{C} \frac{\lambda_1^{n_1} (t - t_1)^{n_1-1} e^{-\lambda_1(t-t_1)}}{(n_1 - 1)!} - a \frac{\lambda_2^{n_2} (t - t_2)^{n_2-1} e^{-\lambda_2(t-t_2)}}{(n_2 - 1)!}$$

where C is the normalizing constant.

- Friston Non-Linear HRF

$$p(t) = \sum_{i=1}^3 \beta_i x_i(t) + \sum_{i=1}^3 \sum_{j=1}^3 \beta_{ij} x_i(t) x_j(t)$$

$$x_i(t) = (r * b_i)(t)$$

$$b_i(t) = \frac{1}{k!} t^k e^{-t} \quad k = 5, 7, 15$$

Experiment steps:

1. initialize parameters
 - (a) number of voxels to simulate
 - (b) exponent of response model
 - (c) double-Gamma HRF
2. Estimate from linear response
3. fit parameter of non-Linear HRF
4. Estimate from non-Linear response
5. repeat 4 to optimize the bar width size
6. repeat 4 with optimized bar width to optimize the rotation angle

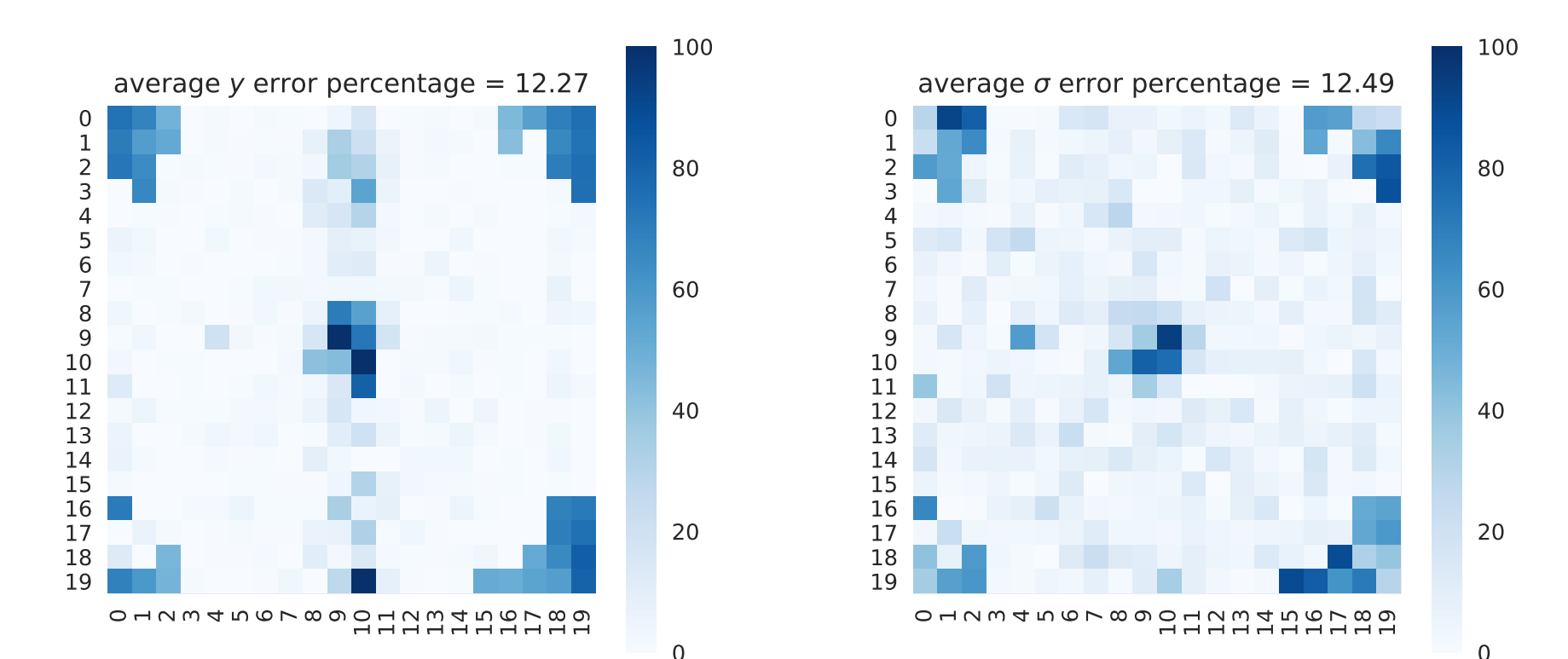
DISCUSSION

We found that non-linearity in simulation HRFs may lead to erroneous pRF estimations. However, we showed that it is possible to optimize the stimulation parameters to ameliorate the effect this non-linearity.

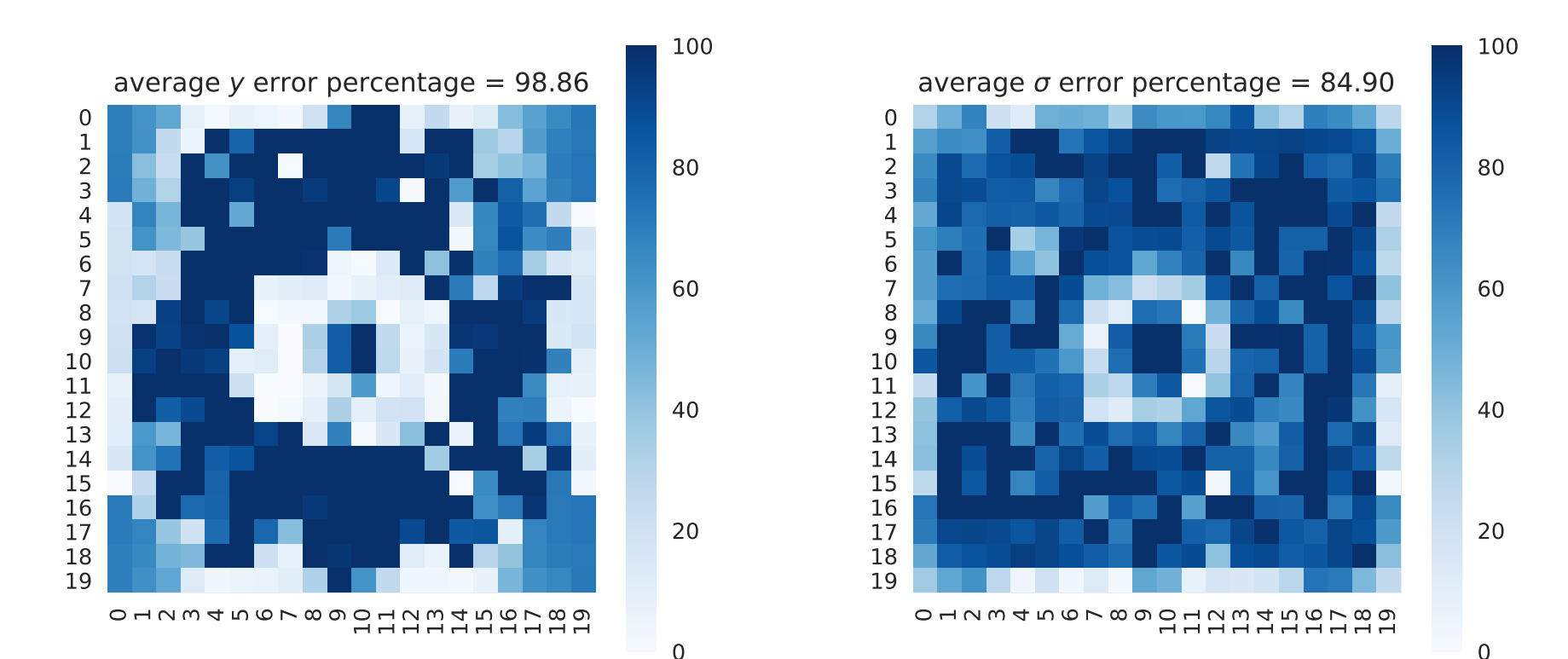
Therefore, we highly recommend that the stimulation protocol (i.e., stimulation and experiment parameters) should be fine-tuned using computer simulations before an actual fMRI experiment is conducted.

RESULTS

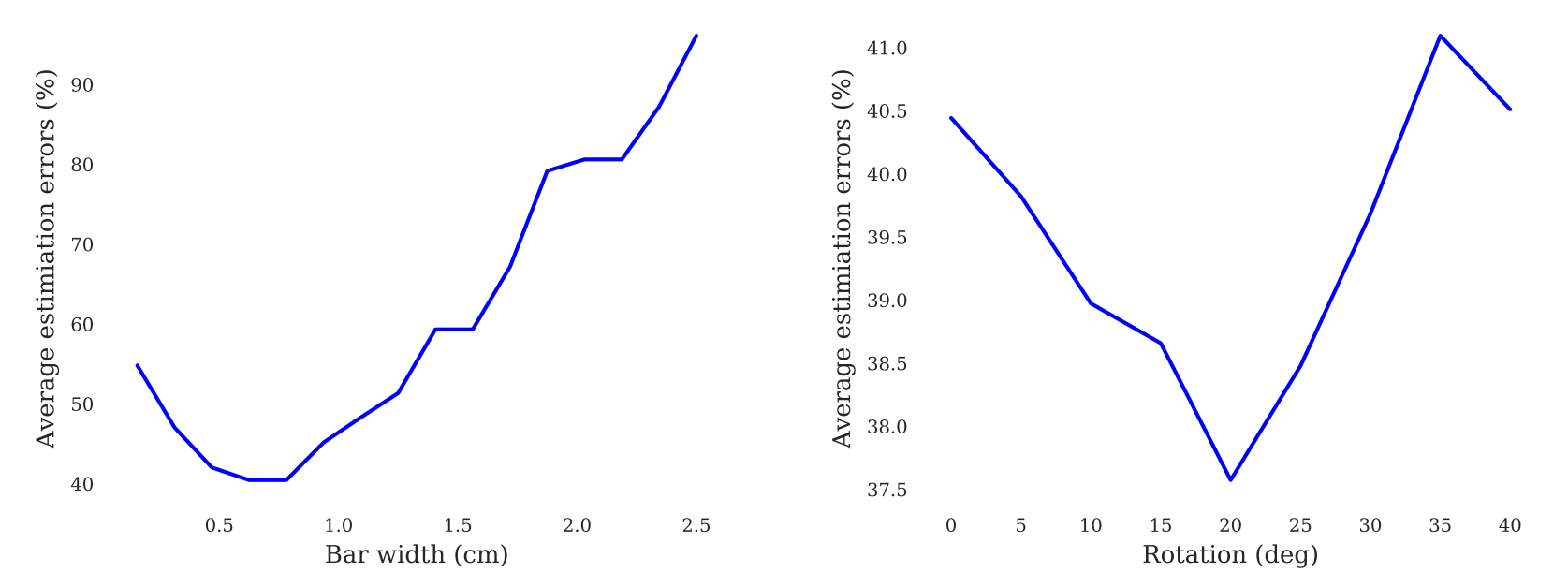
Accuracy maps when simulating with linear HRF and estimating with linear HRF.



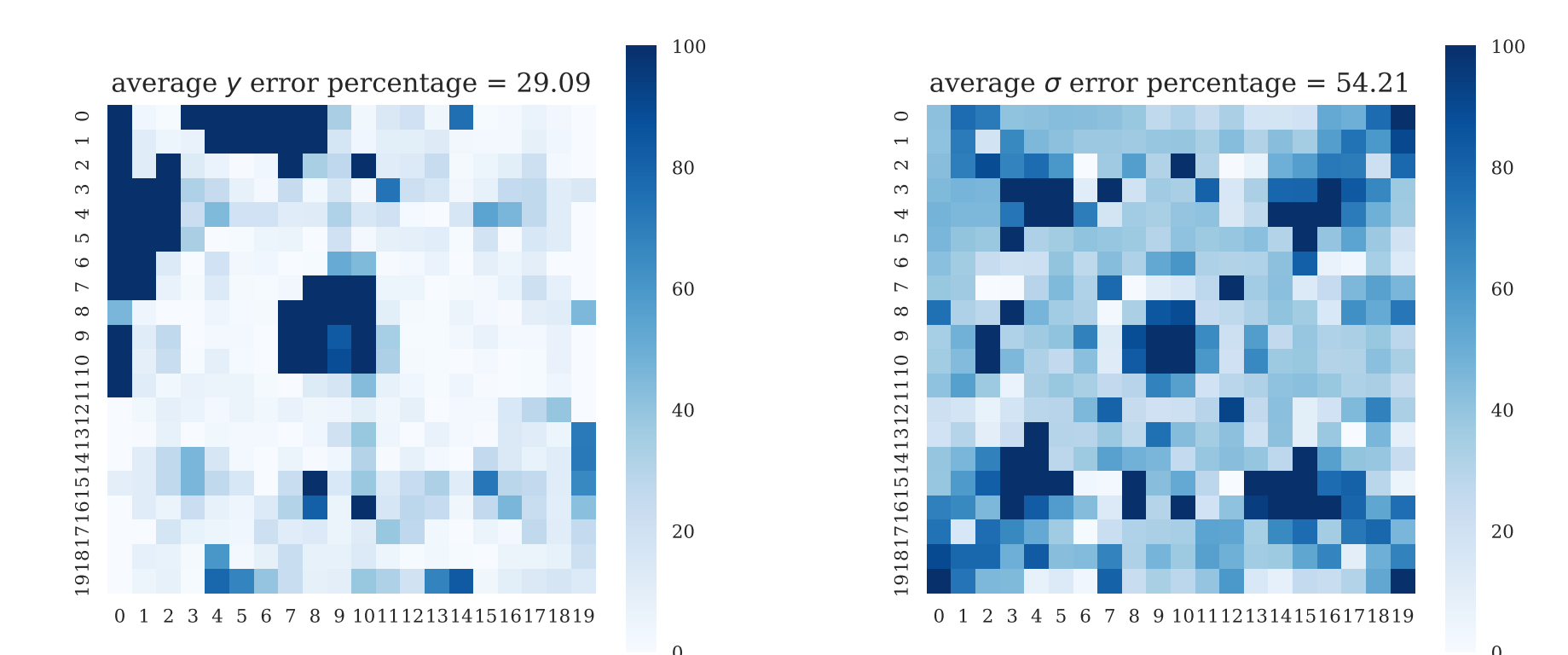
Accuracy maps when simulating with non-linear HRF and estimating with linear HRF:



Optimization of parameters with average of estimation errors over all pRF parameters:



Accuracy maps of same experiment with optimized parameters for size and rotation:



REFERENCES

- [1] S. Dumoulin, B. Wandell. Population Receptive Field Estimates in Human Visual Cortex. In *NeuroImage* '08
- [2] K. Kay and J. Winawer et al. Compressive Spatial Summation in Human Visual Cortex. In *Journal of Neurophysiology* '13

SOURCE CODE

The Python *prfsim* package can directly be installed from PyPI: 'pip install prfsim'
The source code is available at,
github.com/arash-ash/pRFsim

