popeye documentation

Release 0.1

Kevin DeSimone

Jan 10, 2018
Welcome to the documentation for **popeye**!

Contents:
What is a population receptive field?

A population receptive field (pRF) is a quantitative model of the cumulative response of the population of cells contained within a single fMRI voxel. The pRF model can be used to estimate the response properties of populations of neurons using other measures, such as EcOG and EEG.

The pRF model allows us to interpret and predict the responses of a voxel to different stimuli. Such models can be designed to describe various sensory and cognitive processes. More recently, we have used the pRF model to map the retinotopic organization of multiple subcortical nuclei.

Below are parameter maps detailing the response properties of the human lateral geniculate nucleus (LGN). The model parameter estimates are overlaid on the PD image for the left and right LGN. Separate subjects are shown in A and B. Columns illustrate the polar angle, eccentricity, pRF size, and HRF delay estimates. A, anterior; P, posterior.

---

1.1 References
CHAPTER 2

What is popeye?

popeye is an open-source project for developing, estimating, and validating pRF models. popeye was created by Kevin DeSimone with significant guidance and contributions from Ariel Rokem. popeye was developed because there was no open-source software for pRF modeling and estimation. There are several freely available MATLAB toolboxes for doing pRF model estimation, including mrVista and analyzePRF, but since these toolboxes are written in MATLAB they are not truly open-source. This motivated us to create popeye as a project open to critique and collaboration.

The pRF model is what is known as a forward encoding model. We present the participant with a stimulus that varies over time in the tuning dimension(s) of interest. If we’re interested in the retinotopic organization of some brain region, we would use a visual stimulus that varies over the horizontal and vertical dimensions of the visual field. Often experimenters will use a sweeping bar stimulus that makes several passes through the visual field to encode the visuotopic location of the measured response.

The StimulusModel, PopulationModel, and PopulationFit represent the fundamental units of popeye. Each of these are defined in popeye.base. StimulusModel is an abstract class containing the stimulus array—a NumPy ndarray—along with some supporting information about the stimulus timing and encoding. PopulationModel takes a StimulusModel as input and defines a generate_prediction method for operating on the stimulus via some user-defined receptive field model to produce a model prediction given some set of input parameters. PopulationFit takes a PopulationModel and a data timeseries (1D NumPy ndarray), estimating the set of parameters of the PopulationModel that best fits the data.
Welcome to popeye! This section of the documentation will help you install popeye and its dependencies and then walks you through a small demonstration of how to use popeye for fitting some simulated timeseries data.

### 3.1 Installation

Download the popeye source code from the GitHub repository [here](#). Using popeye requires that you have installed NumPy, SciPy, statsmodel, Cython, nibabel, and matplotlib.

Once you’ve downloaded the popeye source code and installed the dependencies, install popeye and build the Cython extensions I’ve written for speeding up the analyses.

```bash
$ cd popeye
$ sudo python setup.py install build_ext
$ python
>>> import popeye
>>> popeye.__version__
'0.1.0.dev'
```

### 3.2 Demo

Below is a small demonstration of how to interact with the popeye API. Here, we’ll generate our stimulus and simulate the BOLD response of a Gaussian pRF model estimate we’ll just invent. Normally, we’d be analyzing the BOLD timeseries that we collect while we present a participant with a visual stimulus.

```python
import ctypes, multiprocessing
import numpy as np
import sharedmem
import popeye.og_hrf as og
import popeye.utilities as utils
```
from popeye.visual_stimulus import VisualStimulus, simulate_bar_stimulus

# seed random number generator so we get the same answers ...
np.random.seed(2764932)

### STIMULUS
## create sweeping bar stimulus
sweeps = np.array([-1,0,90,180,270,-1])  # in degrees, -1 is blank
bar = simulate_bar_stimulus(100, 100, 40, 20, sweeps, 30, 30, 10)

## create an instance of the Stimulus class
stimulus = VisualStimulus(bar, 50, 25, 0.50, 1.0, ctypes.c_int16)

### MODEL
## initialize the gaussian model
model = og.GaussianModel(stimulus, utils.double_gamma_hrf)

## generate a random pRF estimate
x = -5.24
y = 2.58
sigma = 1.24
hrf_delay = -0.25
beta = 0.55
baseline = -0.88

## create the time-series for the invented pRF estimate
data = model.generate_prediction(x, y, sigma, hrf_delay, beta, baseline)

## add some noise
data += np.random.uniform(-data.max()/10, data.max()/10, len(data))

### FIT
## define search grids
# these define min and max of the edge of the initial brute-force search.
x_grid = (-10,10)
y_grid = (-10,10)
s_grid = (1/stimulus.ppd + 0.25,5.25)
b_grid = (0.1,1.0)
h_grid = (-1.0,1.0)

## define search bounds
# these define the boundaries of the final gradient-descent search.
x_bound = (-12.0,12.0)
y_bound = (-12.0,12.0)
s_bound = (1/stimulus.ppd, 12.0)  # smallest sigma is a pixel
b_bound = (1e-8, None)
u_bound = (None, None)
h_bound = (-3.0,3.0)

## package the grids and bounds
grids = (x_grid, y_grid, s_grid, h_grid)
bounds = (x_bound, y_bound, s_bound, h_bound, b_bound, u_bound)

## fit the response
# auto_fit = True fits the model on assignment
# verbose = 0 is silent
# verbose = 1 is a single print
# verbose = 2 is very verbose
3.3 Inspecting the results

Below is the output of the model fit we invoked in the code block above. We also include some matplotlib code for plotting the simulated data and the predicted timeseries. Explore the attributes of the fit object to get a sense of the kinds of measures gleaned from the pRF model.

```python
# plot the results
import matplotlib.pyplot as plt
plt.plot(fit.prediction, c='r', lw=3, label='model', zorder=1)
plt.scatter(range(len(fit.data)), fit.data, s=30, c='k', label='data', zorder=2)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel('Time', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.xlim(0, len(fit.data))
plt.legend(loc=0)
```
CHAPTER 4

API Reference

4.1 popeye

4.2 auditory

- AuditoryFit
- AuditoryModel
- PopulationFit
- PopulationModel
- interp1d
- auto_attr
- fftconvolve
- generate_rf_timeseries_1D
4.2.1 AuditoryFit
4.2.2 AuditoryModel
4.2.3 PopulationFit
4.2.4 PopulationModel
4.2.5 interp1d
4.2.6 auto_attr
4.2.7 fftconvolve
4.2.8 generate_rf_timeseries_1D

4.3 auditory_stimulus

AuditoryStimulus
StimulusModel
auto_attr
generate_spectrogram
imresize
specgram

4.3.1 AuditoryStimulus
4.3.2 StimulusModel
4.3.3 auto_attr
4.3.4 generate_spectrogram
4.3.5 imresize
4.3.6 specgram

4.4 base

PopulationFit
PopulationModel
StimulusModel
auto_attr
set_verbose
4.4.1 PopulationFit
4.4.2 PopulationModel
4.4.3 StimulusModel
4.4.4 auto_attr
4.4.5 set_verbose

4.5 css

CompressiveSpatialSummationFit
CompressiveSpatialSummationModel
PopulationFit
PopulationModel
auto_attr
fftconvolve
generate_og_receptive_field
generate_rf_timeseries

4.5.1 CompressiveSpatialSummationFit
4.5.2 CompressiveSpatialSummationModel
4.5.3 PopulationFit
4.5.4 PopulationModel
4.5.5 auto_attr
4.5.6 fftconvolve
4.5.7 generate_og_receptive_field
4.5.8 generate_rf_timeseries

4.6 dog

DifferenceOfGaussiansFit
DifferenceOfGaussiansModel
PopulationFit
PopulationModel
auto_attr
fftconvolve
generate_og_receptive_field

Continued on next page
Table 4.6 – continued from previous page

<table>
<thead>
<tr>
<th>generate_rf_timeseries</th>
</tr>
</thead>
<tbody>
<tr>
<td>linregress</td>
</tr>
<tr>
<td>trapz</td>
</tr>
</tbody>
</table>

4.6.1 DifferenceOfGaussiansFit

4.6.2 DifferenceOfGaussiansModel

4.6.3 PopulationFit

4.6.4 PopulationModel

4.6.5 auto_attr

4.6.6 fftconvolve

4.6.7 generate_og_receptive_field

4.6.8 generate_rf_timeseries

4.6.9 linregress

4.6.10 trapz

4.7 gabor

<table>
<thead>
<tr>
<th>GaborFit</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaborModel</td>
</tr>
<tr>
<td>PopulationFit</td>
</tr>
<tr>
<td>PopulationModel</td>
</tr>
<tr>
<td>interp1d</td>
</tr>
<tr>
<td>auto_attr</td>
</tr>
<tr>
<td>brute</td>
</tr>
<tr>
<td>fmin_powell</td>
</tr>
<tr>
<td>generate_ballpark_prediction</td>
</tr>
<tr>
<td>linregress</td>
</tr>
</tbody>
</table>
4.7.1 GaborFit
4.7.2 GaborModel
4.7.3 PopulationFit
4.7.4 PopulationModel
4.7.5 interp1d
4.7.6 auto_attr
4.7.7 brute
4.7.8 fmin_powell
4.7.9 generate_ballpark_prediction
4.7.10 linregress

4.8 og

<table>
<thead>
<tr>
<th>GaussianFit</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaussianModel</td>
</tr>
<tr>
<td>PopulationFit</td>
</tr>
<tr>
<td>PopulationModel</td>
</tr>
<tr>
<td>auto_attr</td>
</tr>
<tr>
<td>fftconvolve</td>
</tr>
<tr>
<td>generate_og_receptive_field</td>
</tr>
<tr>
<td>generate_rf_timeseries</td>
</tr>
</tbody>
</table>
4.8.1 GaussianFit
4.8.2 GaussianModel
4.8.3 PopulationFit
4.8.4 PopulationModel
4.8.5 auto_attr
4.8.6 fftconvolve
4.8.7 generate_og_receptive_field
4.8.8 generate_rf_timeseries

<table>
<thead>
<tr>
<th>4.9 ogb</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaussianFit</td>
</tr>
<tr>
<td>GaussianModel</td>
</tr>
<tr>
<td>PopulationFit</td>
</tr>
<tr>
<td>PopulationModel</td>
</tr>
<tr>
<td>auto_attr</td>
</tr>
<tr>
<td>fftconvolve</td>
</tr>
<tr>
<td>generate_og_receptive_field</td>
</tr>
<tr>
<td>generate_rf_timeseries</td>
</tr>
</tbody>
</table>

4.9.1 GaussianFit
4.9.2 GaussianModel
4.9.3 PopulationFit
4.9.4 PopulationModel
4.9.5 auto_attr
4.9.6 fftconvolve
4.9.7 generate_og_receptive_field
4.9.8 generate_rf_timeseries

4.10 onetime

| OneTimeProperty |
Table 4.10 – continued from previous page

<table>
<thead>
<tr>
<th>ResetMixin</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_attr</td>
</tr>
<tr>
<td>setattr_on_read</td>
</tr>
</tbody>
</table>

4.10.1 OneTimeProperty

4.10.2 ResetMixin

4.10.3 auto_attr

4.10.4 setattr_on_read

4.11 plotting

<table>
<thead>
<tr>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFormatterMathtext</td>
</tr>
<tr>
<td>LogNorm</td>
</tr>
<tr>
<td>gaussian_kde</td>
</tr>
<tr>
<td>beta_hist</td>
</tr>
<tr>
<td>eccentricity_hist</td>
</tr>
<tr>
<td>eccentricity_sigma_fill</td>
</tr>
<tr>
<td>eccentricity_sigma_scatter</td>
</tr>
<tr>
<td>field_coverage</td>
</tr>
<tr>
<td>find</td>
</tr>
<tr>
<td>generate_og_receptive_field</td>
</tr>
<tr>
<td>hexbin_location_map</td>
</tr>
<tr>
<td>hrf_delay_hist</td>
</tr>
<tr>
<td>hrf_delay_kde</td>
</tr>
<tr>
<td>lazy_field_coverage</td>
</tr>
<tr>
<td>linregress</td>
</tr>
<tr>
<td>location_and_size_map</td>
</tr>
<tr>
<td>location_estimate_jointdist</td>
</tr>
<tr>
<td>make_movie_calleable</td>
</tr>
<tr>
<td>make_movie_static</td>
</tr>
<tr>
<td>polar_angle_plot</td>
</tr>
<tr>
<td>sigma_hrf_delay_scatter</td>
</tr>
</tbody>
</table>
4.11.1 Circle
4.11.2 LogFormatterMathtext
4.11.3 LogNorm
4.11.4 gaussian_kde
4.11.5 beta_hist
4.11.6 eccentricity_hist
4.11.7 eccentricity_sigma_fill
4.11.8 eccentricity_sigma_scatter
4.11.9 field_coverage
4.11.10 find
4.11.11 generate_og_receptive_field
4.11.12 hexbin_location_map
4.11.13 hrf_delay_hist
4.11.14 hrf_delay_kde
4.11.15 lazy_field_coverage
4.11.16 linregress
4.11.17 location_and_size_map
4.11.18 location_estimate_jointdist
4.11.19 make_movie_calleable
4.11.20 make_movie_static
4.11.21 polar_angle_plot
4.11.22 sigma_hrf_delay_scatter

4.12 reconstruction

---

Process
Continued on next page
<table>
<thead>
<tr>
<th>interp1d</th>
<th>Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>find</td>
<td>A multi-producer, multi-consumer queue.</td>
</tr>
<tr>
<td>multiprocess_stimulus_reconstruction</td>
<td></td>
</tr>
<tr>
<td>multiprocess_stimulus_reconstruction_realtime</td>
<td></td>
</tr>
<tr>
<td>reconstruct_stimulus</td>
<td></td>
</tr>
<tr>
<td>reconstruct_stimulus_realtime</td>
<td></td>
</tr>
<tr>
<td>reconstruct_stimulus_realtime_smoothing</td>
<td></td>
</tr>
</tbody>
</table>

### 4.12.1 Process

### 4.12.2 interp1d

### 4.12.3 Array

### 4.12.4 Queue

### 4.12.5 find

### 4.12.6 multiprocess_stimulus_reconstruction

### 4.12.7 multiprocess_stimulus_reconstruction_realtime

### 4.12.8 reconstruct_stimulus

### 4.12.9 reconstruct_stimulus_realtime

### 4.12.10 reconstruct_stimulus_realtime_smoothing

### 4.13 simulation

<table>
<thead>
<tr>
<th>VisualStimulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeat</td>
</tr>
<tr>
<td>error_function</td>
</tr>
<tr>
<td>fmin</td>
</tr>
<tr>
<td>fmin_powell</td>
</tr>
<tr>
<td>generate_og_receptive_field</td>
</tr>
<tr>
<td>generate_og_receptive_fields</td>
</tr>
<tr>
<td>generate_scatter_volume</td>
</tr>
<tr>
<td>parallel_simulate_neural_sigma</td>
</tr>
<tr>
<td>recast_simulation_results</td>
</tr>
<tr>
<td>resample_stimulus</td>
</tr>
<tr>
<td>shuffle</td>
</tr>
<tr>
<td>simulate_bar_stimulus</td>
</tr>
<tr>
<td>simulate_neural_sigma</td>
</tr>
</tbody>
</table>
4.13.1 VisualStimulus

4.13.2 repeat

4.13.3 error_function

4.13.4 fmin

4.13.5 fmin_powell

4.13.6 generate_og_receptive_field

4.13.7 generate_og_receptive_fields

4.13.8 generate_scatter_volume

4.13.9 parallel_simulate_neural_sigma

4.13.10 recast_simulation_results

4.13.11 resample_stimulus

4.13.12 shuffle

4.13.13 simulate_bar_stimulus

4.13.14 simulate_neural_sigma

4.14 spectrotemporal

```
PopulationFit
PopulationModel
SpectroTemporalFit
SpectroTemporalModel
interp1d
auto_attr
brute
compute_model_ts
decimate
fftconvolve
fmin_powell
gaussian_1D
gaussian_2D
generate_rf_timeseries_1D
generate_strf_timeseries
imresize
linregress
median_filter
```

Continued on next page
Table 4.14 – continued from previous page

<table>
<thead>
<tr>
<th>parallel_fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>recast_estimation_results</td>
</tr>
<tr>
<td>romb</td>
</tr>
<tr>
<td>trapz</td>
</tr>
</tbody>
</table>
4.14.1 PopulationFit
4.14.2 PopulationModel
4.14.3 SpectroTemporalFit
4.14.4 SpectroTemporalModel
4.14.5 interp1d
4.14.6 auto_attr
4.14.7 brute
4.14.8 compute_model_ts
4.14.9 decimate
4.14.10 fftconvolve
4.14.11 fmin_powell
4.14.12 gaussian_1D
4.14.13 gaussian_2D
4.14.14 generate_rf_timeseries_1D
4.14.15 generate_strf_timeseries
4.14.16 imresize
4.14.17 linregress
4.14.18 median_filter
4.14.19 parallel_fit
4.14.20 recast_estimation_results
4.14.21 romb
4.14.22 trapz
4.15 utilities

repeat

Continued on next page
Table 4.15 – continued from previous page

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
</tr>
<tr>
<td>brute</td>
</tr>
<tr>
<td>brute_force_search</td>
</tr>
<tr>
<td>double_gamma_hrf</td>
</tr>
<tr>
<td>error_function</td>
</tr>
<tr>
<td>fmin</td>
</tr>
<tr>
<td>fmin_powell</td>
</tr>
<tr>
<td>gaussian_2D</td>
</tr>
<tr>
<td>generate_shared_array</td>
</tr>
<tr>
<td>gradient_descent_search</td>
</tr>
<tr>
<td>imresize</td>
</tr>
<tr>
<td>make_nifti</td>
</tr>
<tr>
<td>multiprocess_bundle</td>
</tr>
<tr>
<td>normalize</td>
</tr>
<tr>
<td>parallel_fit</td>
</tr>
<tr>
<td>percent_change</td>
</tr>
<tr>
<td>recast_estimation_results</td>
</tr>
<tr>
<td>romb</td>
</tr>
<tr>
<td>trapz</td>
</tr>
<tr>
<td>zscore</td>
</tr>
</tbody>
</table>
4.15.1 repeat
4.15.2 Array
4.15.3 brute
4.15.4 brute_force_search
4.15.5 double_gamma_hrf
4.15.6 error_function
4.15.7 fmin
4.15.8 fmin_powell
4.15.9 gaussian_2D
4.15.10 generate_shared_array
4.15.11 gradient_descent_search
4.15.12 imresize
4.15.13 make_nifti
4.15.14 multiprocess_bundle
4.15.15 normalize
4.15.16 parallel_fit
4.15.17 percent_change
4.15.18 recast_estimation_results
4.15.19 romb
4.15.20 trapz
4.15.21 zscore

4.16 version
4.17 visual_stimulus

StimulusModel
VisualStimulus
gaussian_2D
generate_coordinate_matrices
imread
imresize
loadmat
resample_stimulus
simulate_bar_stimulus
simulate_checkerboard_bar
simulate_movie_bar
simulate_sinflicker_bar
square
zoom

4.17.1 StimulusModel

4.17.2 VisualStimulus

4.17.3 gaussian_2D

4.17.4 generate_coordinate_matrices

4.17.5 imread

4.17.6 imresize

4.17.7 loadmat

4.17.8 resample_stimulus

4.17.9 simulate_bar_stimulus

4.17.10 simulate_checkerboard_bar

4.17.11 simulate_movie_bar

4.17.12 simulate_sinflicker_bar

4.17.13 square

4.17.14 zoom

4.18 xvalidation
4.18.1 coeff_of_determination

4.18.2 deepcopy

4.18.3 kfold_xval

4.18.4 parallel_xval

4.18.5 recast_xval_results
CHAPTER 5

Indices and tables

• genindex
• modindex
• search