Script to plot Figure 3bc

Table of Contents

Winawer, Kay, Foster, Parvizi, Wandell **Asynchronous broadband signals are the principal source of the BOLD response in human visual cortex** *Current Biology, 2013*

This figure shows the prf centers (panel b) and prf compressive exponent (panel c) from fitting the CSS model (Compressive Spatial Summation) to both the asynchronous broadband and stimulus-locked time series

Copyright Jonathan Winawer, 2013

A note on reproducibility: The plots produced from this script differ very slightly from the plots in Figure 3 in the publication. This is due to a slight difference in the stimulus descriptions used for solving the pRF models. In this script, the stimuli used as inputs for the pRF models are binary masks. For the publication the stimuli were floats approximating a binary mask, with occasional pixel values differing slightly from 0 or 1 due to an imperfection in the algorithm that converted the image indices used for the experiments into into binary contrast masks.

Set up paths and parameters

```
% Path to save the eps figures
savepth   = fullfile(ecogPRFrootPath, 'scratch');

% Paths to load the pre-computed AB and SL pRF model solutions
abPRFfile = fullfile(ecogPRFrootPath, 'data', 'PRF_exp_ab');
slPRFfile = fullfile(ecogPRFrootPath, 'data', 'PRF_exp_sl');
```

Load the PRF model solutions

The pRF models are pre-solved. If you would like to resolve them, run the following script:

```
        s_ecogSolvePRFs

% Load the two sets of PRF model solutions
prf.ab = load(abPRFfile);
prf.sl = load(slPRFfile);
```

Select appropriate channels

For summarizing population data, we impose the following selection criteria:

1. Channels are within ROIs V1/V2/V3
2. Variance explained from both AB and SL pRF models exceeds 30%
3. Experiments were conducted with flickering patterns rather than static patterns (which means, include subjects 1:3, exclude S4)

For further details, see supplementary table 1 and supplementary methods section 'Channel Selection'.

```
abok = find(prf.ab.params.isV1V2V3 & prf.ab.params.subj < 4 & prf.ab.params.r > 30
slok = find(prf.sl.params.isV1V2V3 & prf.sl.params.subj < 4 & prf.sl.params.r > 30
okchannels = intersect(abok, slok);
```

Figure 3b: Scatterplot of PRF centers

```
fH = figure;set(fH, 'Color', 'w');

stimulusExtent = 10; % degrees

set(gca, 'FontSize', 18,'ColorOrder', jet(length(okchannels)),  ...
    'XTick',  [-1 0 1]* stimulusExtent, 'YTick', [-1 0 1]* stimulusExtent);

xlabel('Visual field position (deg)')

hold all

% Plot a circle to denote the maximum stimulus extent
th = linspace(0, 2*pi, 30);
aperture.x = cos(th) * stimulusExtent;
aperture.y = sin(th) * stimulusExtent;

% Get the x,y center positions of for the AB and SL pRF models
xab = prf.ab.params.x(okchannels);
yab = prf.ab.params.y(okchannels);
xsl = prf.sl.params.x(okchannels);
ysl = prf.sl.params.y(okchannels);

% Line segments to connect the AB and SL pRF centers from a given channel
plot([xab xsl]', [yab ysl]', '-k', 'LineWidth',1.5);

% Plot the pRF centers
for ii = 1:length(xsl); plot(xsl(ii), ysl(ii), 'o', 'MarkerSize', 12, ...
        'MarkerFaceColor', [.2 .2 .2], 'MarkerEdgeColor', 'k'); end
for ii = 1:length(xab); plot(xab(ii), yab(ii), 's', 'MarkerSize', 12, ...
        'MarkerFaceColor', [1 1 1], 'MarkerEdgeColor', [0 0 0]); end

% Add some grid lines
plot([0 0], [-1 1]*stimulusExtent, 'k-', [-1 1]*stimulusExtent, [0 0], 'k-')
plot(aperture.x, aperture.y, 'k-')

% Limit the axes to the stimulus extent and squarify
axis([[-1 1]*stimulusExtent [-1 1]*stimulusExtent])
axis square
```

```
title(sprintf('PRF centers\ncircles: stimulus-locked; squares: broadband'))
```

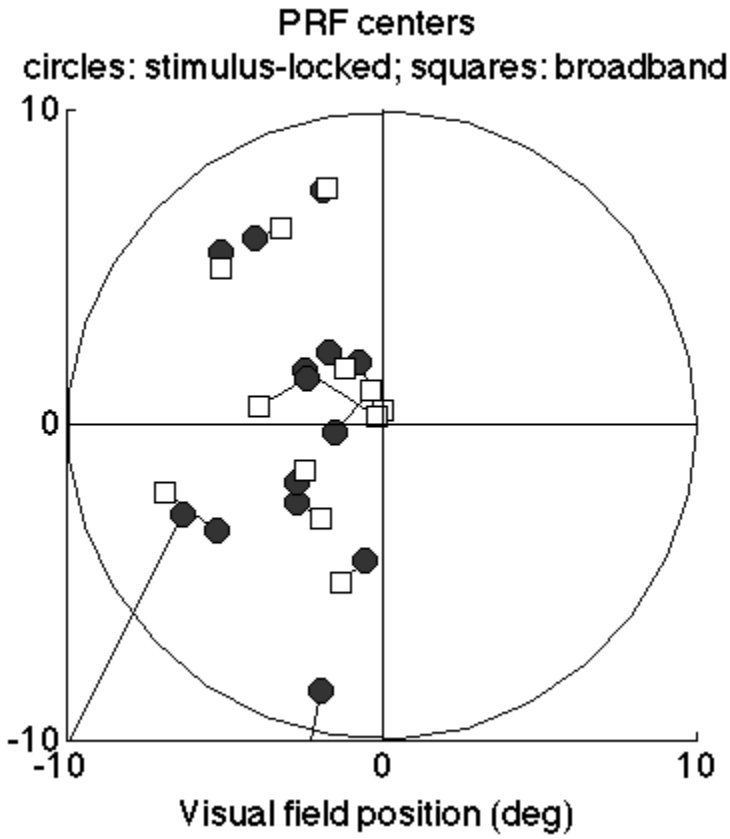
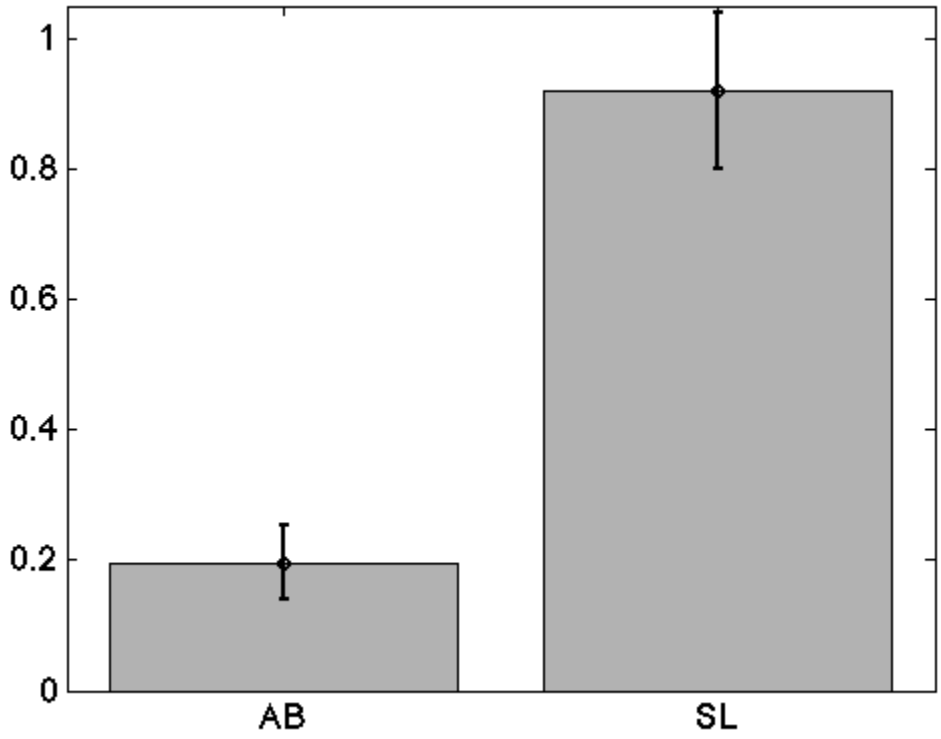


Figure 3c: Bar plot of PRF exponents

```
% Note fMRI data not re-plotted. See Figure 3c in text for fMRI data.

% Mean pRF exponents for AB and SL pRF models
mn = [];
mn.ab = mean(prf.ab.params.params(okchannels,5));
mn.sl = mean(prf.sl.params.params(okchannels,5));

% Standard error of the pRF exponents for AB and SL pRF models
se.ab = std(prf.ab.params.params(okchannels,5))/sqrt(length(okchannels));
se.sl = std(prf.sl.params.params(okchannels,5))/sqrt(length(okchannels));

% Make a bar plot
fH(2) = figure; set(fH(2), 'Color', 'w'); set(gca, 'FontSize', 18);

bar([1 2], [mn.ab mn.sl], 'FaceColor', [.7 .7 .7]); hold on
errorbar([1 2], [mn.ab mn.sl], [se.ab se.sl], 'ok', 'LineWidth', 2);
set(gca, 'XTick', [1 2], 'XTickLabel', {'AB', 'SL'}, 'YLim', [0 1.05])

% %% Save 'em
%
% hgexport(fH(1), fullfile(savepth, 'Figure3b_AB_and_SL_pRF_Centers.eps'));
% hgexport(fH(2), fullfile(savepth, 'Figure3c_AB_and_SL_pRF_Exponents.eps'));
```

```
%
% return
```



Published with MATLAB® 8.0