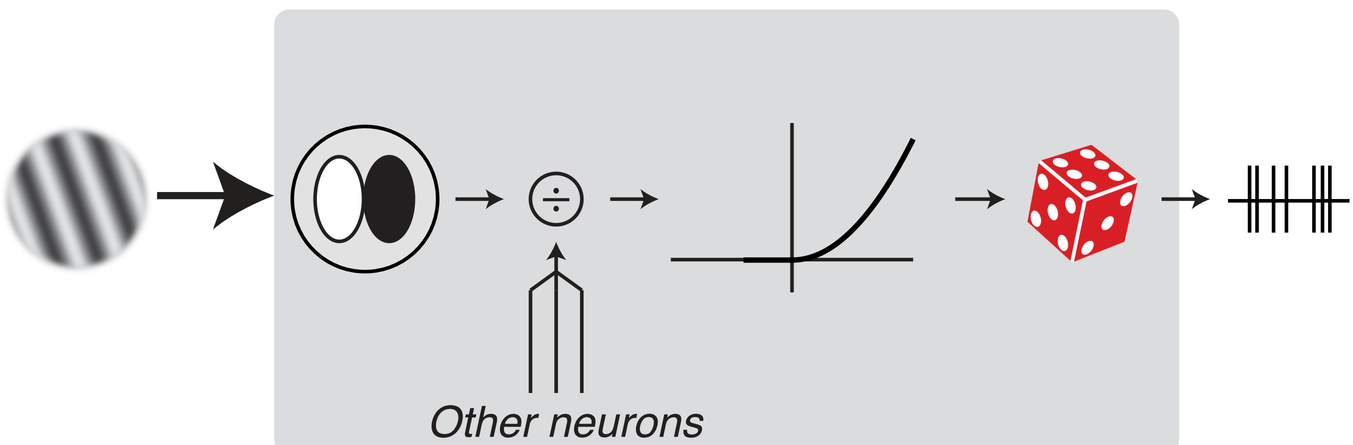
Modeling spatial frequency selectivity in V1
11.02.17

Goal

Our recordings in macaque V1 have expanded upon the observation that spatial frequency preference and bandwidth are contrast dependent. Specifically, by presenting stimuli of varying spectral dispersion - rather than simple sinusoidal gratings alone - we have shown that the systematic shift of spatial frequency preference towards lower spatial frequencies with reduced contrast is enhanced with increased dispersion. To better understand which computations might support the spatial frequency tuning V1 neurons, we now aim to build a computational model of spatial frequency processing in cortex. By fitting model parameters for each cell, we can then identify which model components (i.e. calculations/computations) give rise to the observed "behavior" of each cell.

Form of the model



On a coarse level, the model receives input in the form of the properties of the image components (e.g. the contrast or spatial frequency of each grating) and computes a linear filter response which is scaled by a contrast gain control and passed through a response nonlinearity to obtain a firing rate. The optimization routine seeks to minimize an objective function based on some loss measurement, for each trial, between the measured neural response and the model prediction. Thus far, the model has fared modestly well at capturing the overall shape of spatial frequency selectivity and changes in response amplitude with manipulations in contrast and dispersion, but has failed to replicate changes in spatial frequency preference with contrast.

Optimized parameters:
Linear filter: The linear filtering is accomplished by a derivative-of-Gaussian kernel.
1. **Preferred spatial frequency (in c/deg)**
2. **Derivative order applied to the Gaussian filter**

Gain control: A divisive normalization serves as gain control. The properties of the neurons which comprise the normalization pool are not fit parameters but rather have been chosen by hand. There are two pools of neurons which span the space of spatial frequencies - one with broader filters and another with narrower filters (and a comparatively higher gain). Linear filter responses in quadrature are thresholded and squared before being superimposed. The final normalization signal, then, is determined by two parameters: a constant normalization strength and a weighting of the normalization channels.

3. **Stimulus-independent constant term** (added to computed response)
4. **Asymmetry of normalization weighting** (currently we can "tilt" the weighting of normalization responses - i.e. we can change the slope of weights over spatial frequency to be stronger either at high or low spatial frequencies)

Response weighting and thresholding:

5. **Response exponent** (i.e. steepness of response nonlinearity)
6. **Response scalar** (i.e. overall magnitude of response - applied after normalization, nonlinearity)

Noise:

7. **"Early" noise** (added to excitatory response of linear filter; using "early" rather than "late" noise allows for the response to be suppressed below baseline)
8. **"Late" noise** (added to mean response after all other calculations - stimulus independent baseline)

Currently not used:

9. **Aspect ratio** (used to control the selectivity of orientation tuning)
10. **Orientation preference**
11. **Direction selectivity**
12. **Subtractive gain** (used either as a constant subtractive term or as a gain on the normalization signal which is also applied as a subtractive term on the excitatory filter response)
13. **Variation of gain** (determines the variance of the spike count; used in the log-likelihood loss calculation)

Calculating the response:

Model responses are computed in the Fourier domain on a trial-by-trial and frame-by-frame basis. The linear filter and normalization pool receive the same information: the orientation, spatial frequency, temporal frequency, and contrast of all gratings present in the stimulus (recall that each stimulus is made up of 9 gratings, though many have zero contrast in the lower dispersion cases). The selectivity of the filter to the stimulus is determined by the filter parameters, namely the derivative order and preferred spatial frequency. This selectivity is multiplied by the contrast of each grating, the result of which is multiplied by the filter response after accounting for the phase of each grating. For the linear filter, this response is then half-wave rectified (i.e. thresholded at zero); for the normalization pool responses, the responses are half-wave rectified and then squared. These initial normalization responses are then scaled according to the asymmetry parameter (see parameter 4 above); the square root of this value is taken after the appropriate gain is applied to bring things back to linear.

The overall response is then computed in the following way. The numerator $a = \eta_e + R_{lf}$ - i.e. the sum of the early noise (parameter 7) and the linear filter response. The denominator b is $b = \sqrt{\sigma^2 + \gamma^2}$ where σ is the constant normalization term (parameter 3 above) and γ is the gain-adjusted normalization response described at the end of the above paragraph. Then, the response $r = max(0, \frac{a}{b})^{\alpha}$ where a, b are as above and α is the response exponent. All of

r, a, b are computed for each frame, and thus a trial-by-trial response is computed by $\tilde{r} = \eta_l + \beta * \mu(r)$ with η_l as late noise (parameter 8), β as the response gain (parameter 6), and $\mu(r)$ the mean over all frames of the response computed above for each trial.

Loss functions:
Thus far, we have worked with two loss functions.

- We started with a negative-log likelihood calculation where the likelihood was determined by the negative binomial distribution. In short, we compare the number of "successful observations" (the number of observed spikes) to a rate and overall number of "observations" based on the model predicted rate and variance.
- We have since moved to computing a loss which is similar to mean squared error but instead relies on squaring the difference of the square roots of the relevant quantities (model rate versus observed rate) rather than simply the quantities unaltered. That is, $l := (\sqrt{r_m} - \sqrt{r_n})^2$ where "m" indicates model response and "n" indicates recorded neural response.

Importantly, we made the change from the former loss function to the latter one because we felt the model was allowing too much "leeway" for high-response conditions. The second loss function described will more greatly penalize an equivalent model miss for a condition with a high response rate than will the first loss function.

Optimization:
Very early versions of the model were coded in Matlab and optimized using fmincon - i.e. constrained optimization in Matlab using one of the standard optimization algorithms. Since roughly mid-April (2017), optimization has been performed in python using the TensorFlow module. The optimization routine used is the Adam optimizer, which is based on first-order stochastic gradient descent but - among other advantages - more carefully adapts the step-size of each parameter independently. Until 9/25/17, optimization was stopped after a fixed number of iterations; now, the optimization runs until the change in the loss function is less than some bound or the allotted time on the NYU HPC (high performance computing) cluster is exhausted. The former case is far more common.

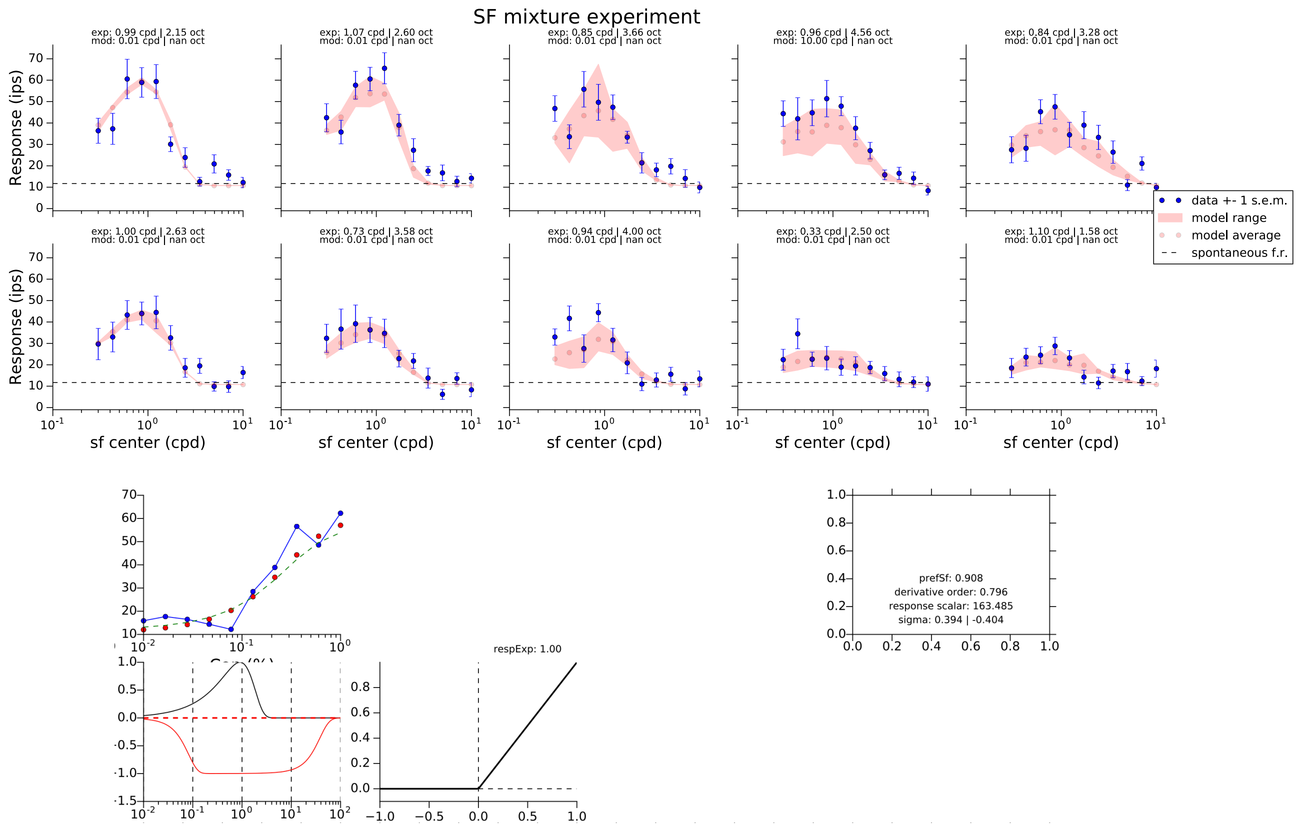
State of the model

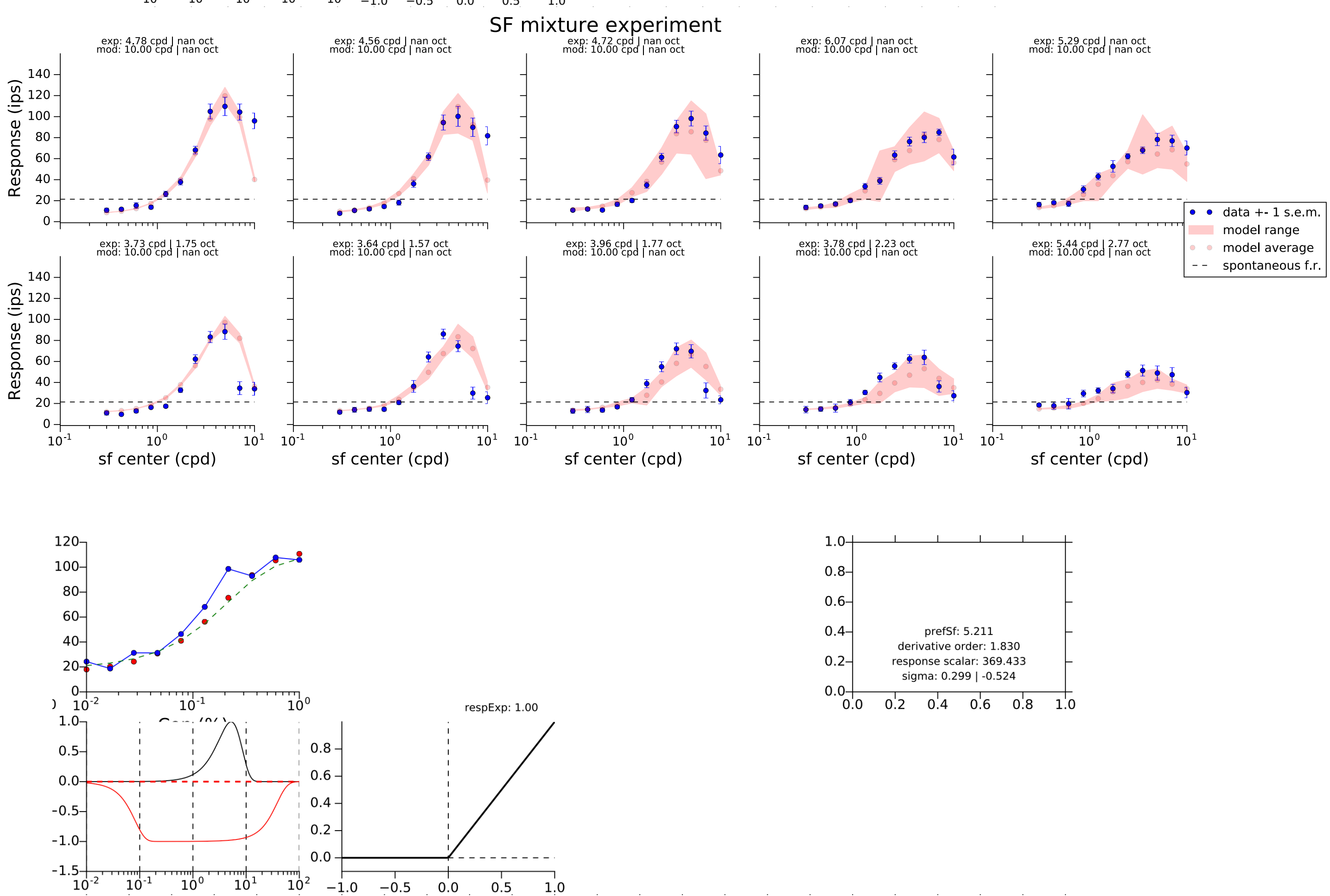
I have explored numerous iterations of the model - adding or dropping parameters to be fit, changing the bounds of the parameters, altering the loss function, etc. The first eight parameters described above are in the current form of the model, and the loss function is the altered least squared error term. As described above, current work is done in TensorFlow (Python) using the Adam optimization routine with termination once changes in the loss function (measured on all data for that cell every 500 steps) are less than some criterion.

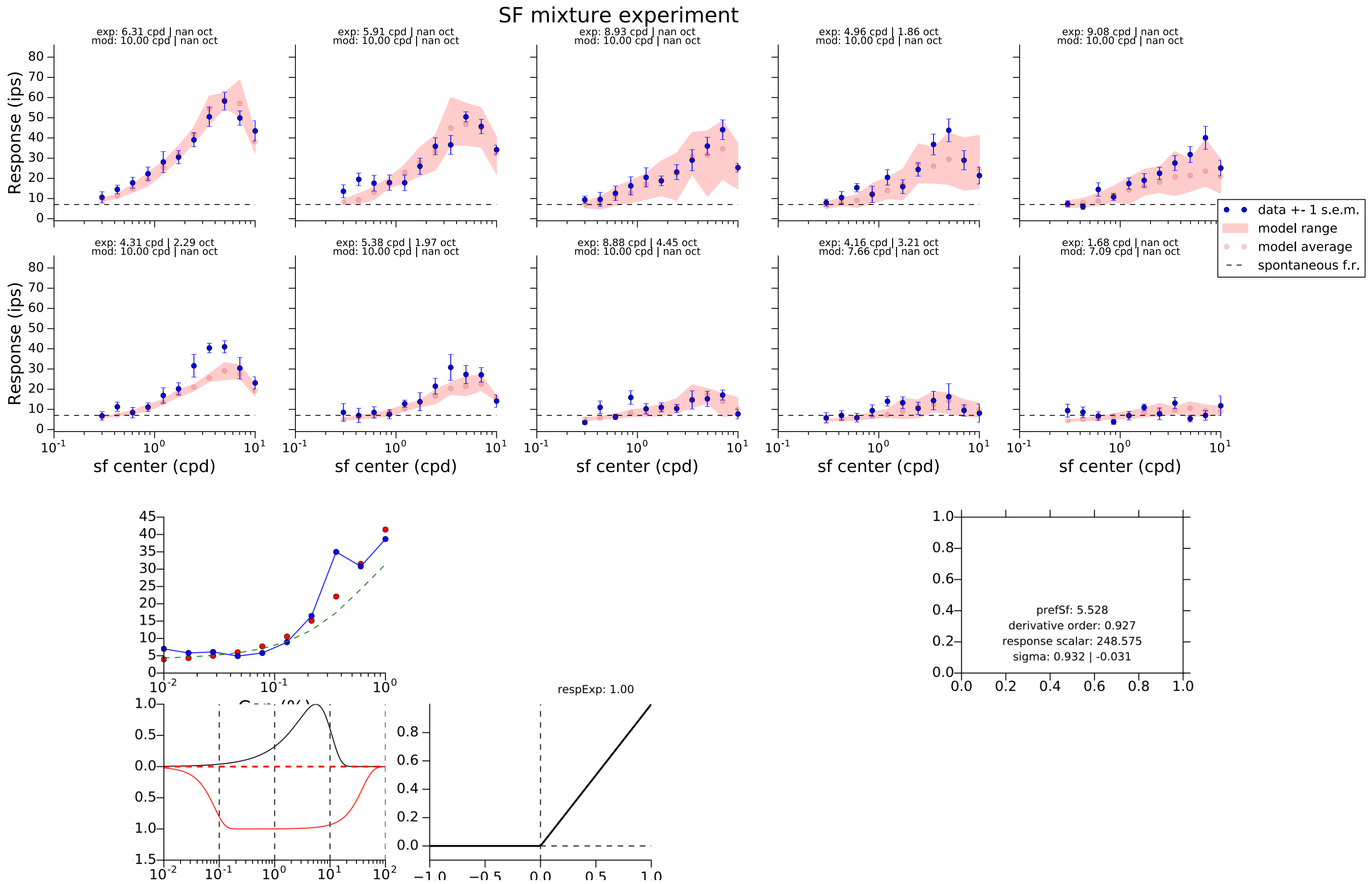
Overall, the model fits are somewhat respectable, but problems still exist. There is a tendency for the model to overestimate the response at high contrast/low dispersion conditions and also to underestimate the response at low contrast/high dispersion conditions, though recent model fits have improved upon this. We also know that model is not capable of capturing the shift in preferred spatial frequency that occur with changes in contrast.

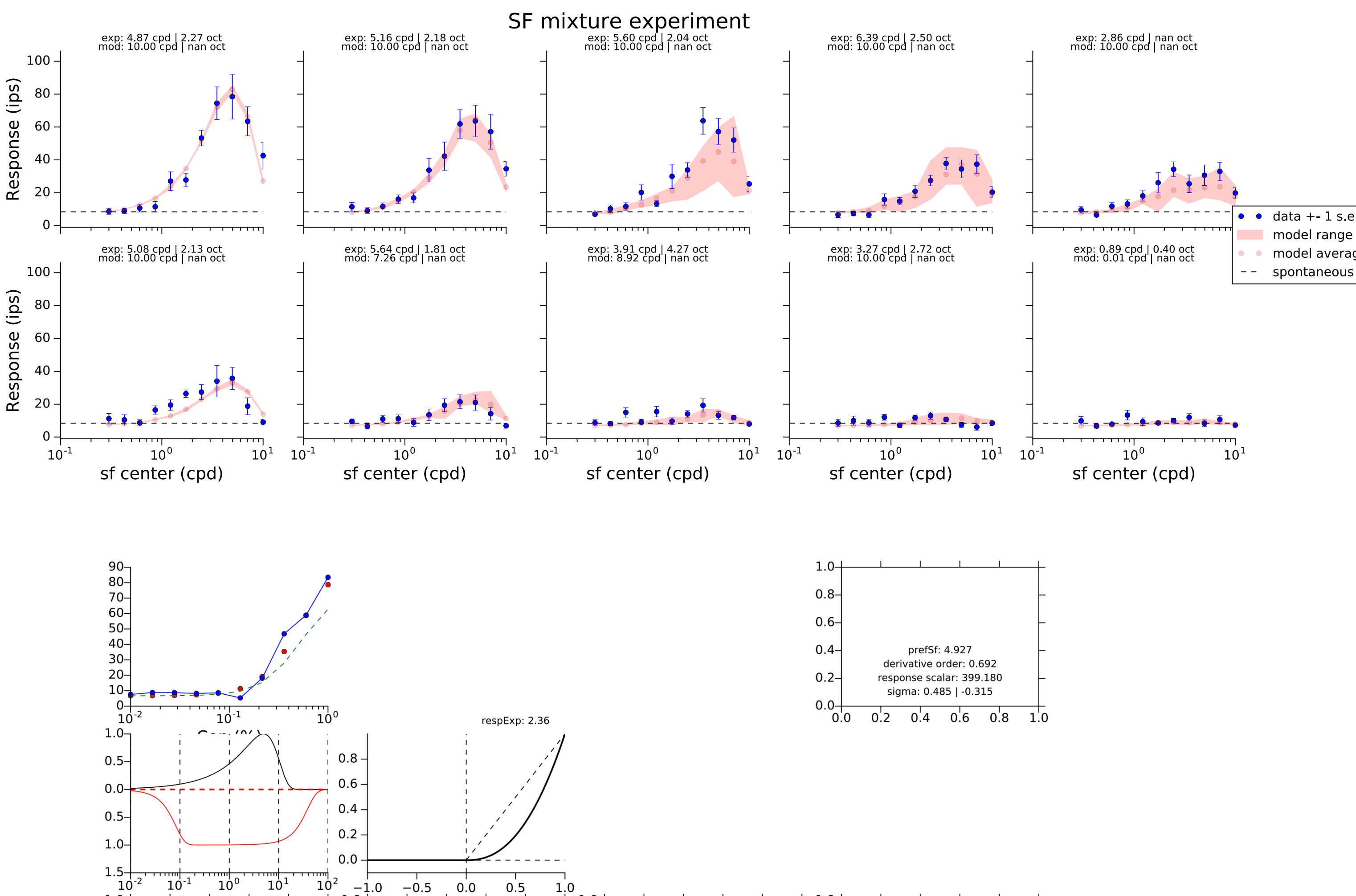
To highlight the current state of the model, and to readily compare to previous model iterations, we'll highlight a few example cell responses and their model fits. All examples will have - in order - cells 2, 27, 47, 57

Current fits

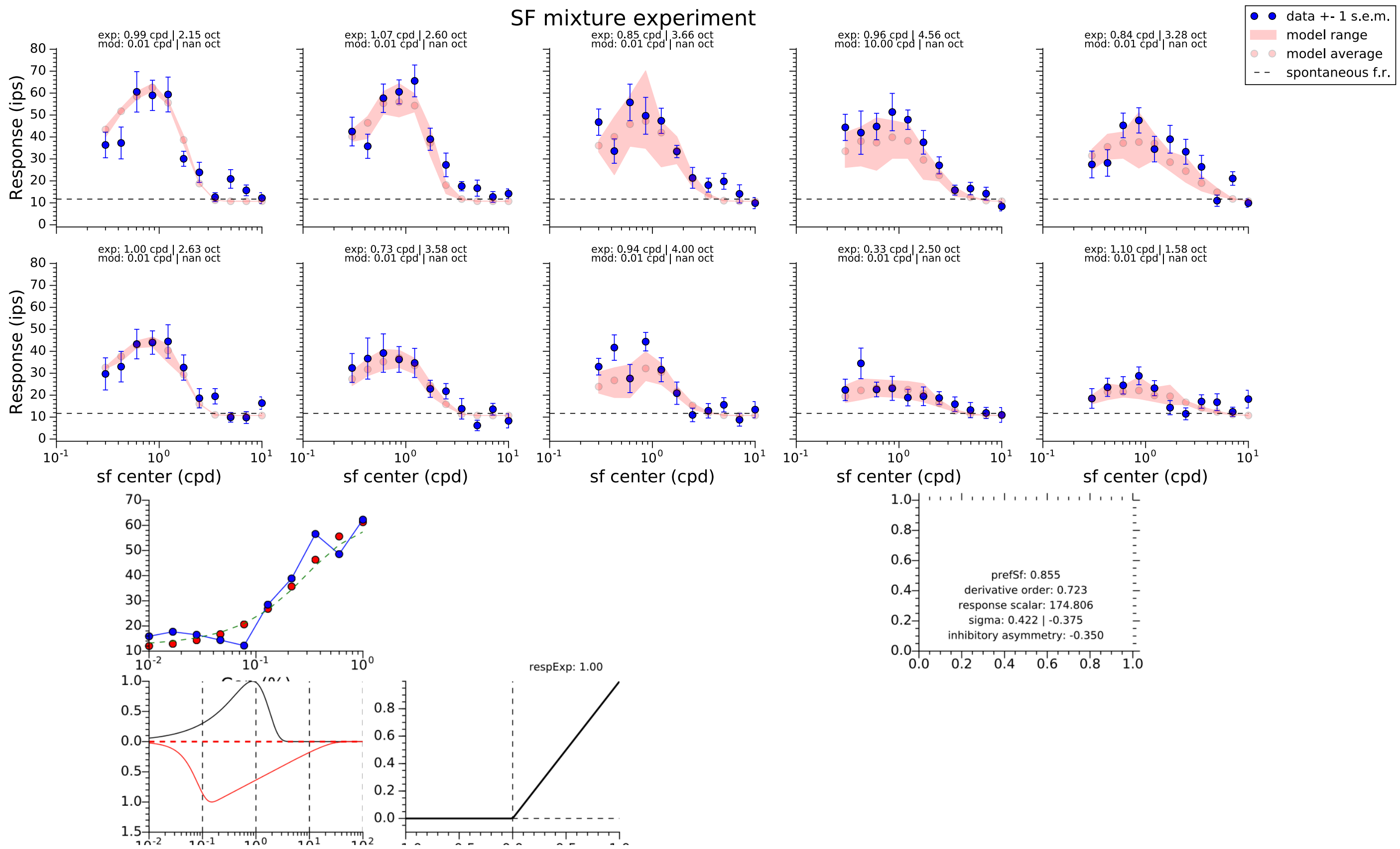


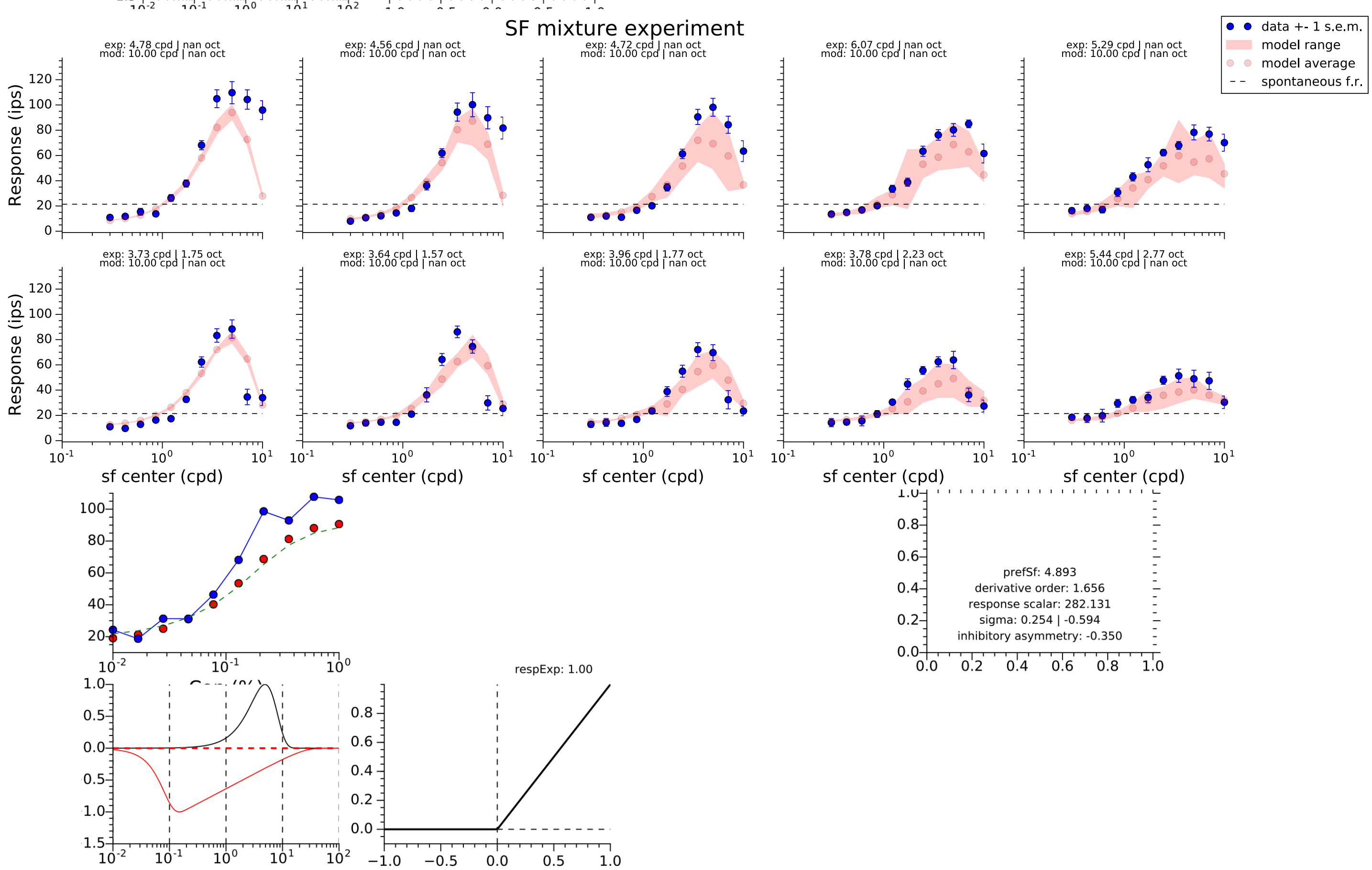


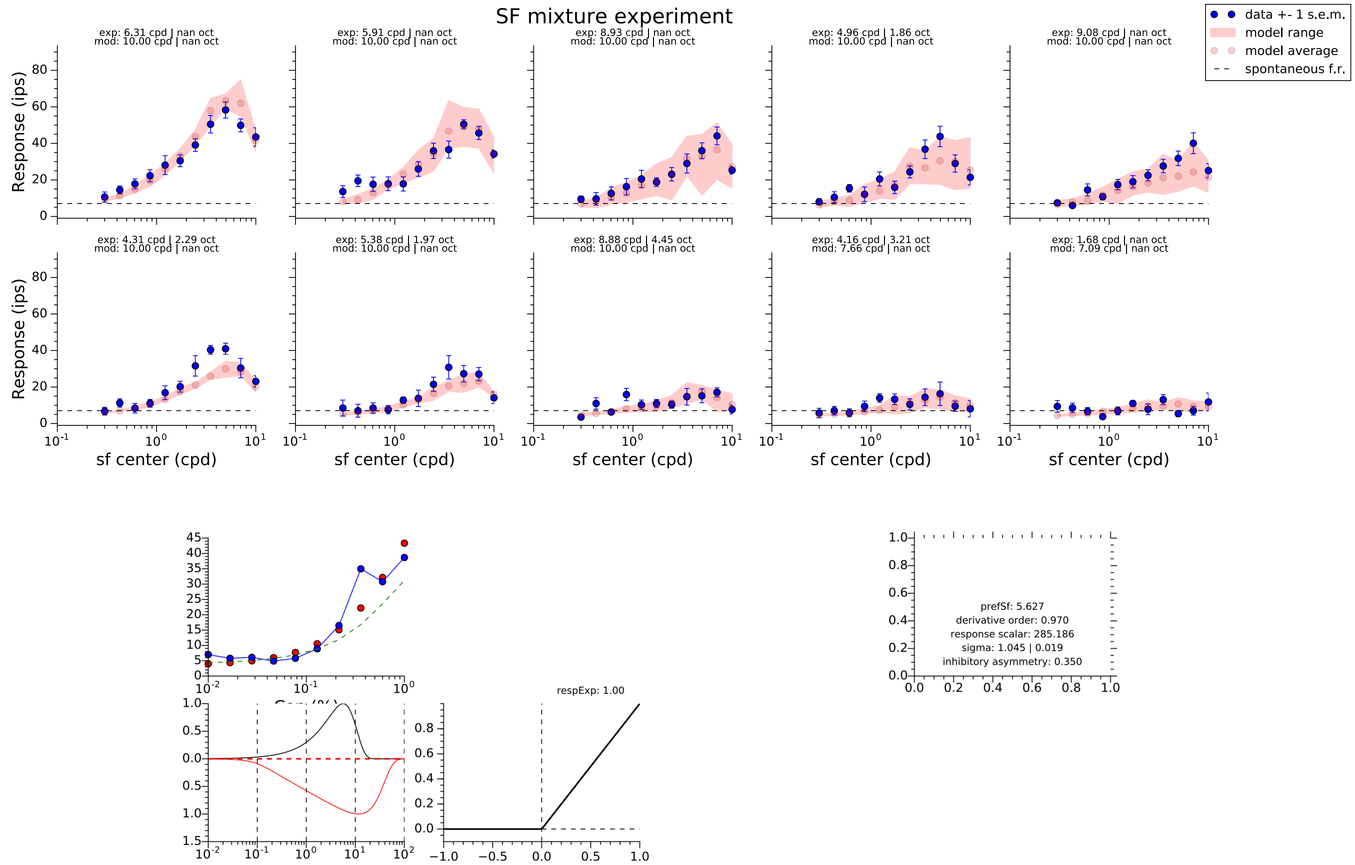

SF mixture experiment
exp: 6.31 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.91 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 8.93 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.96 cpd | 1.86 oct
mod: 10.00 cpd | nan oct
exp: 9.08 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.31 cpd | 2.29 oct
mod: 10.00 cpd | nan oct
exp: 5.38 cpd | 1.97 oct
mod: 10.00 cpd | nan oct
exp: 8.88 cpd | 4.45 oct
mod: 10.00 cpd | nan oct
exp: 4.16 cpd | 3.21 oct
mod: 7.66 cpd | nan oct
exp: 1.68 cpd | nan oct
mod: 7.09 cpd | nan oct
Response (ips)
sf center (cpd)
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
prefSf: 5.528
derivative order: 0.927
response scalar: 248.575
sigma: 0.932 | -0.031
respExp: 1.00

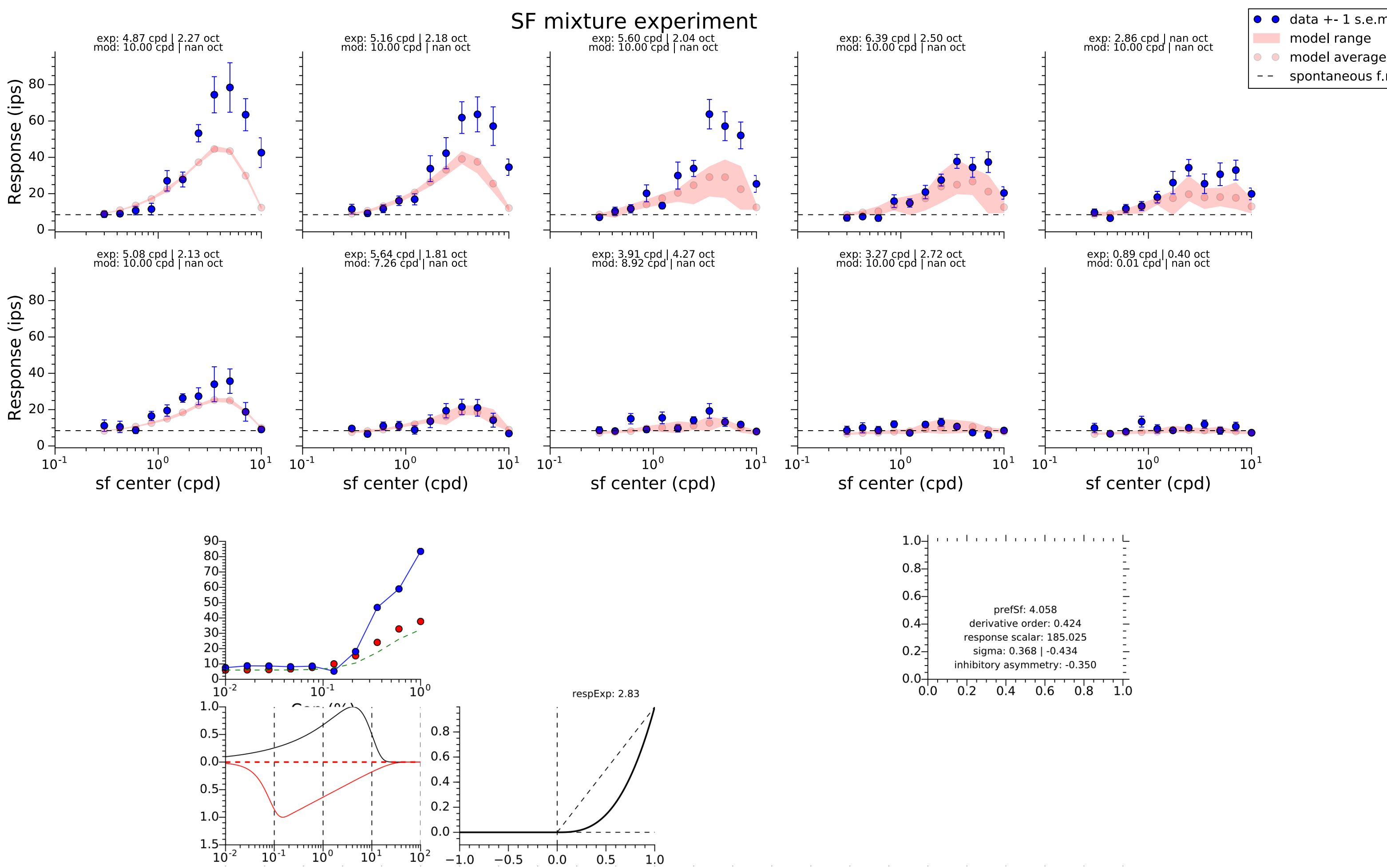

SF mixture experiment
exp: 4.87 cpd | 2.27 oct
mod: 10.00 cpd | nan oct
exp: 5.16 cpd | 2.18 oct
mod: 10.00 cpd | nan oct
exp: 5.60 cpd | 2.04 oct
mod: 10.00 cpd | nan oct
exp: 6.39 cpd | 2.50 oct
mod: 10.00 cpd | nan oct
exp: 2.86 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.08 cpd | 2.13 oct
mod: 10.00 cpd | nan oct
exp: 5.64 cpd | 1.81 oct
mod: 7.26 cpd | nan oct
exp: 3.91 cpd | 4.27 oct
mod: 8.92 cpd | nan oct
exp: 3.27 cpd | 2.72 oct
mod: 10.00 cpd | nan oct
exp: 0.89 cpd | 0.40 oct
mod: 0.01 cpd | nan oct
Response (ips)
sf center (cpd)
data +- 1 s.e.m.
model range
prefSf: 4.927
derivative order: 0.692
response scalar: 399.180
sigma: 0.485 | -0.315
respExp: 2.36

Fits with asymmetric normalization

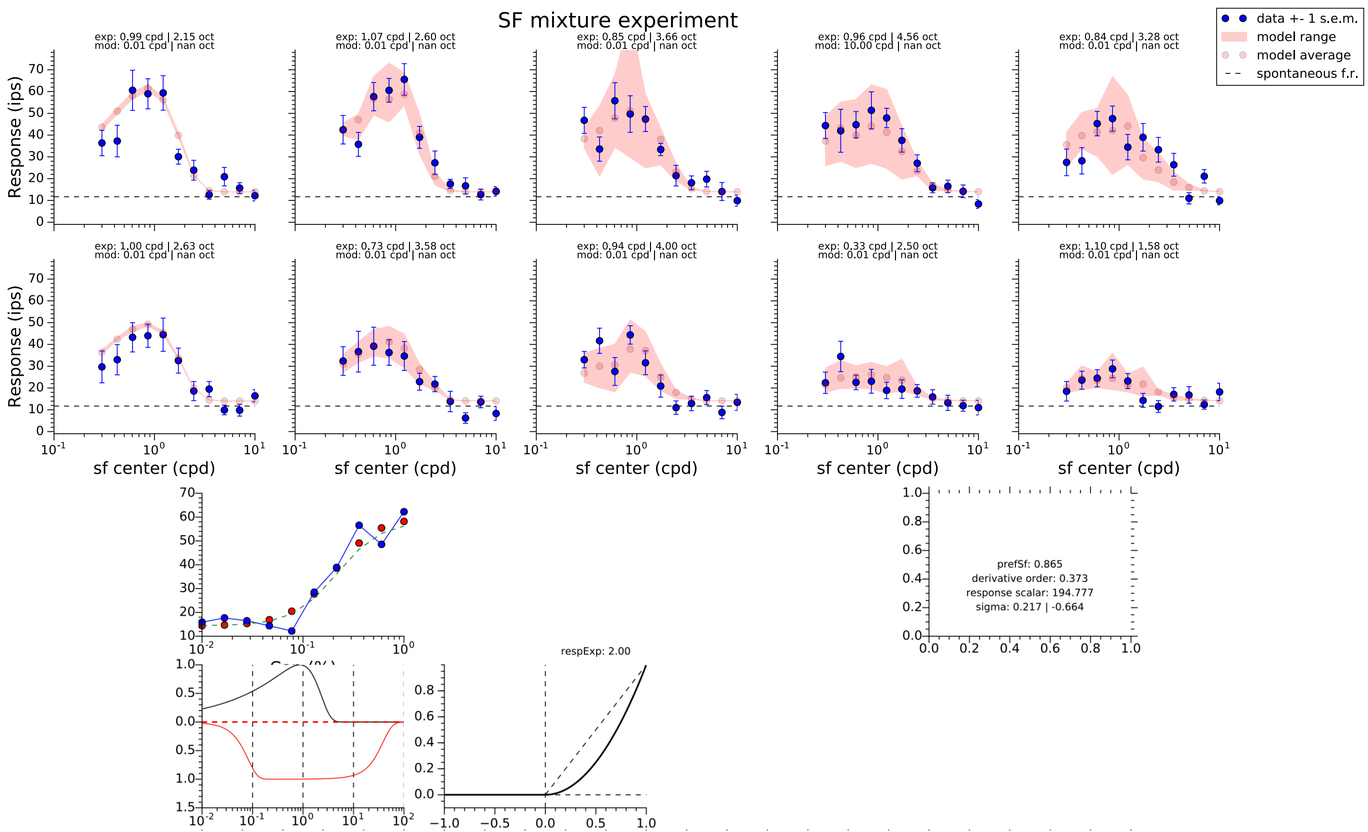


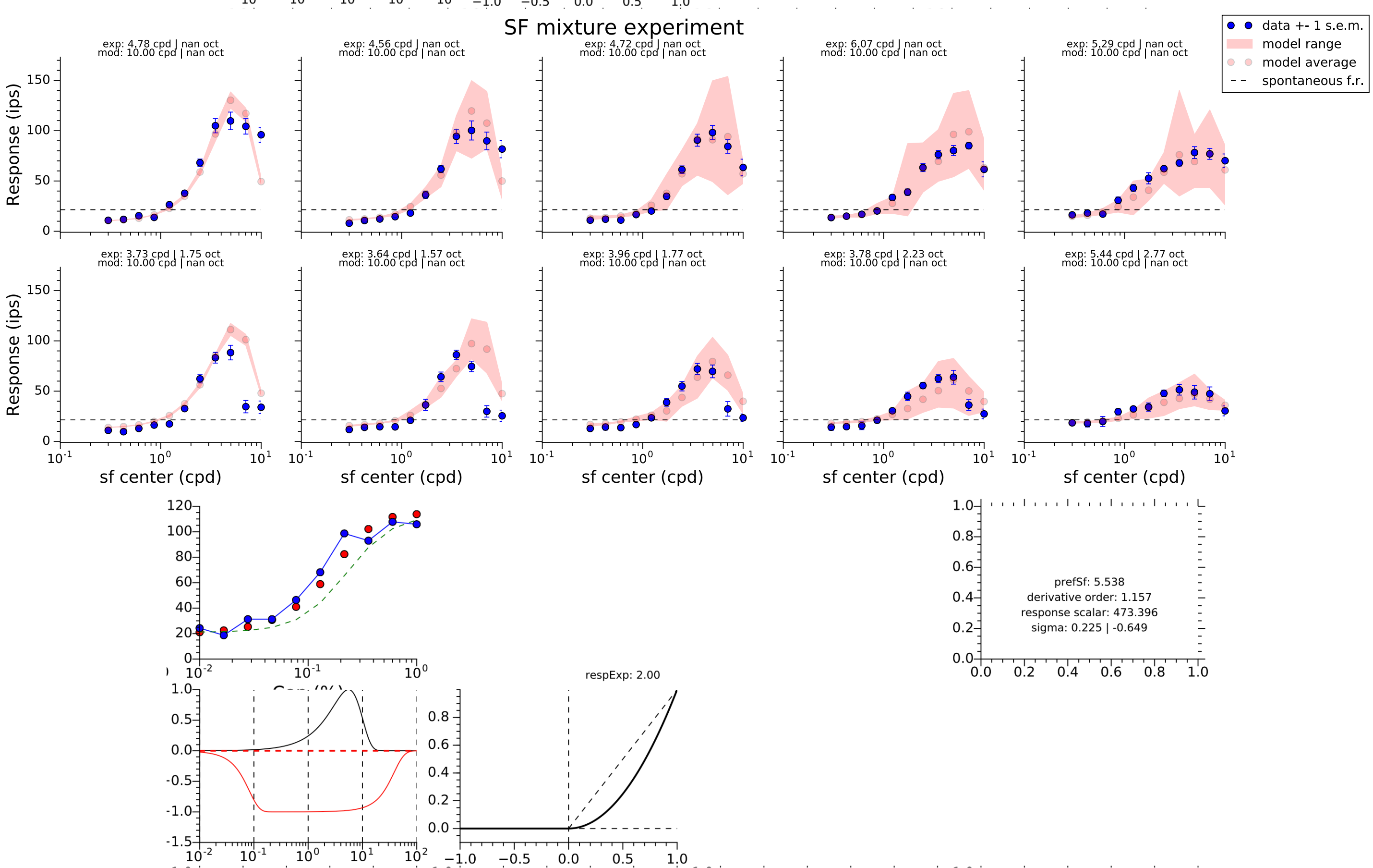


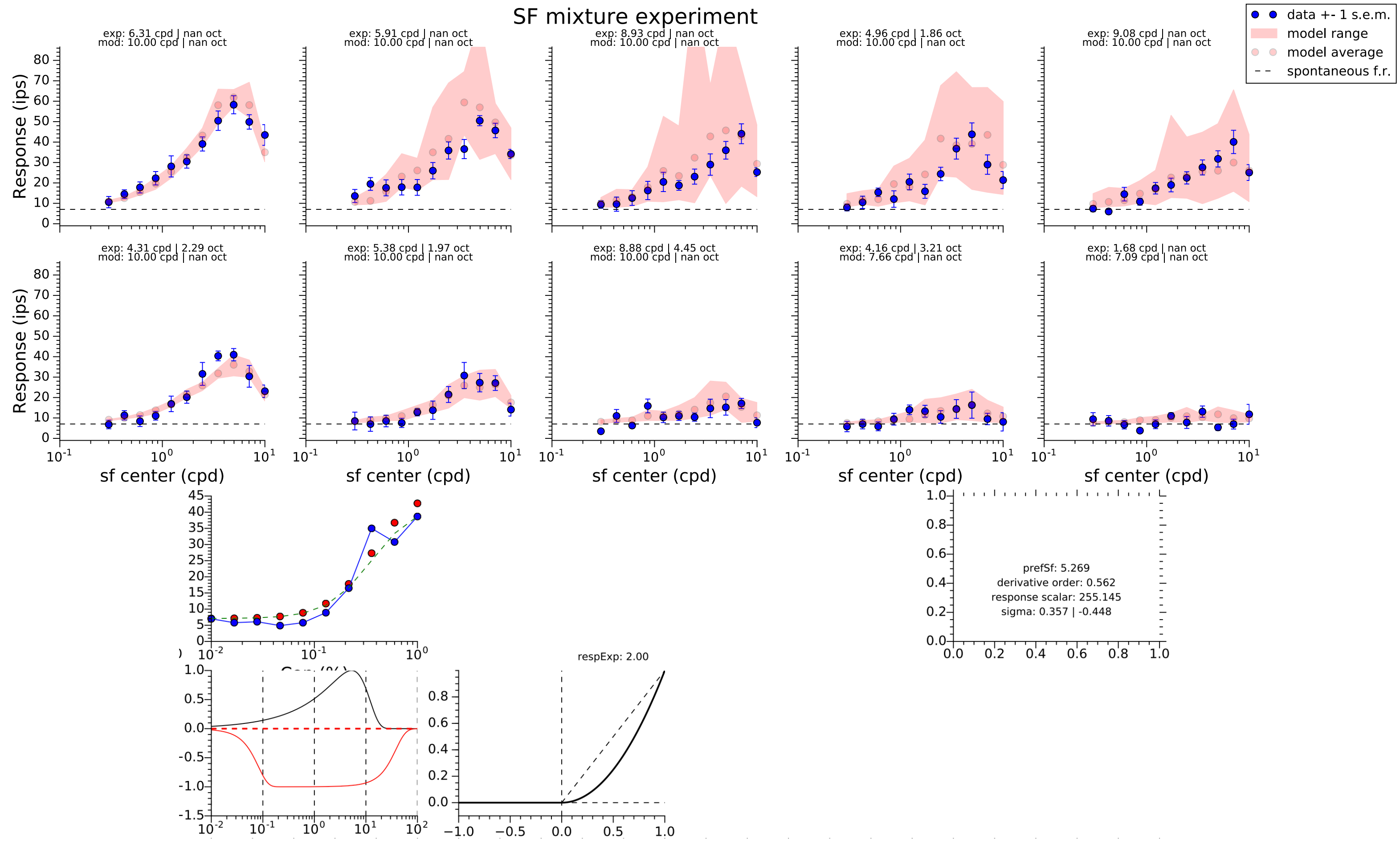

SF mixture experiment
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
exp: 6.31 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.91 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 8.93 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.96 cpd | 1.86 oct
mod: 10.00 cpd | nan oct
exp: 9.08 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.31 cpd | 2.29 oct
mod: 10.00 cpd | nan oct
exp: 5.38 cpd | 1.97 oct
mod: 10.00 cpd | nan oct
exp: 8.88 cpd | 4.45 oct
mod: 10.00 cpd | nan oct
exp: 4.16 cpd | 3.21 oct
mod: 7.66 cpd | nan oct
exp: 1.68 cpd | nan oct
mod: 7.09 cpd | nan oct
Response (ips)
sf center (cpd)
prefSf: 5.627
derivative order: 0.970
response scalar: 285.186
sigma: 1.045 | 0.019
inhibitory asymmetry: 0.350
respExp: 1.00

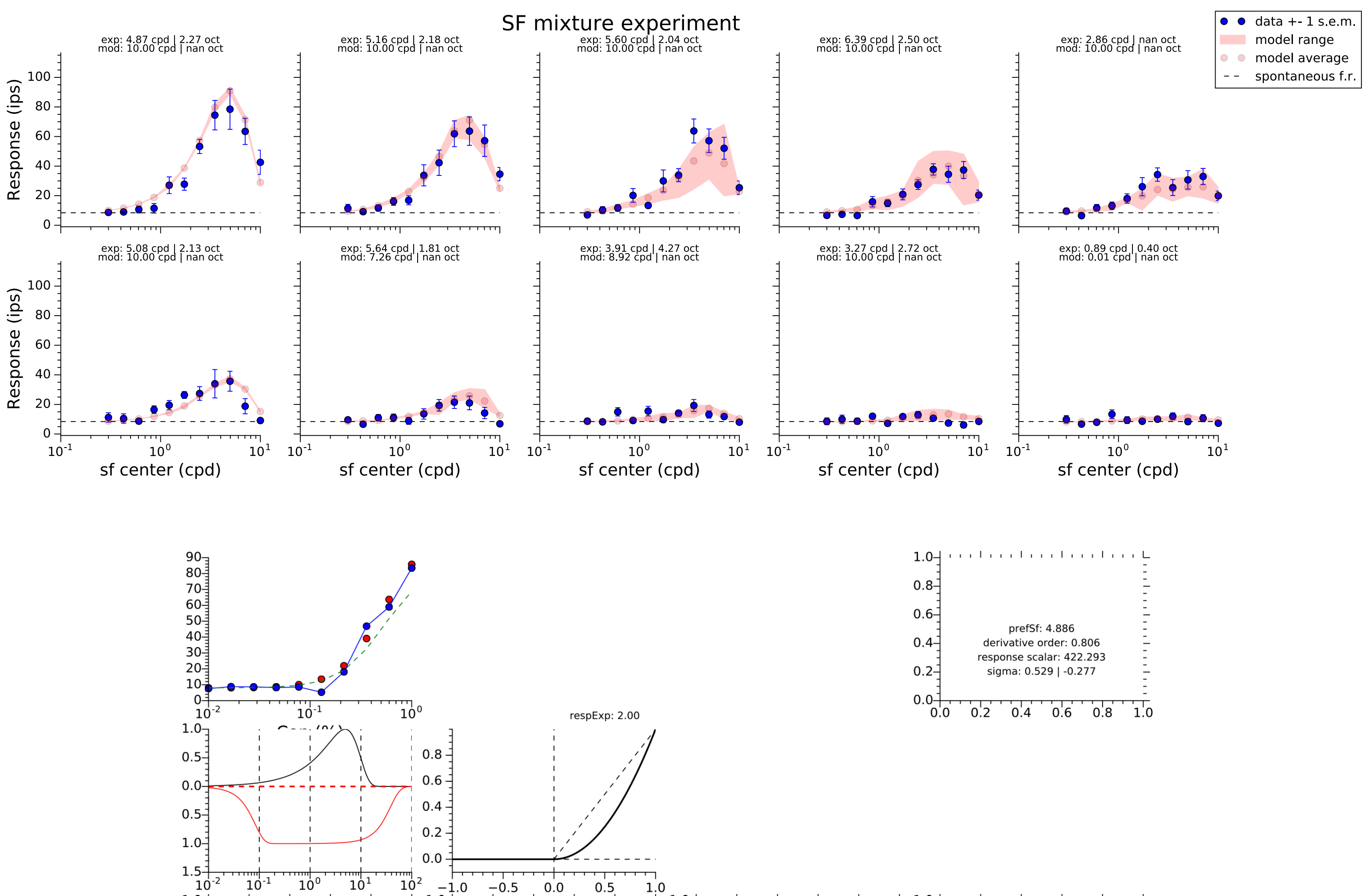

SF mixture experiment
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
exp: 4.87 cpd | 2.27 oct
mod: 10.00 cpd | nan oct
exp: 5.16 cpd | 2.18 oct
mod: 10.00 cpd | nan oct
exp: 5.60 cpd | 2.04 oct
mod: 10.00 cpd | nan oct
exp: 6.39 cpd | 2.50 oct
mod: 10.00 cpd | nan oct
exp: 2.86 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.08 cpd | 2.13 oct
mod: 10.00 cpd | nan oct
exp: 5.64 cpd | 1.81 oct
mod: 7.26 cpd | nan oct
exp: 3.91 cpd | 4.27 oct
mod: 8.92 cpd | nan oct
exp: 3.27 cpd | 2.72 oct
mod: 10.00 cpd | nan oct
exp: 0.89 cpd | 0.40 oct
mod: 0.01 cpd | nan oct
Response (ips)
sf center (cpd)
prefSf: 4.058
derivative order: 0.424
response scalar: 185.025
sigma: 0.368 | -0.434
inhibitory asymmetry: -0.350
respExp: 2.83

Response exponent fixed at 2

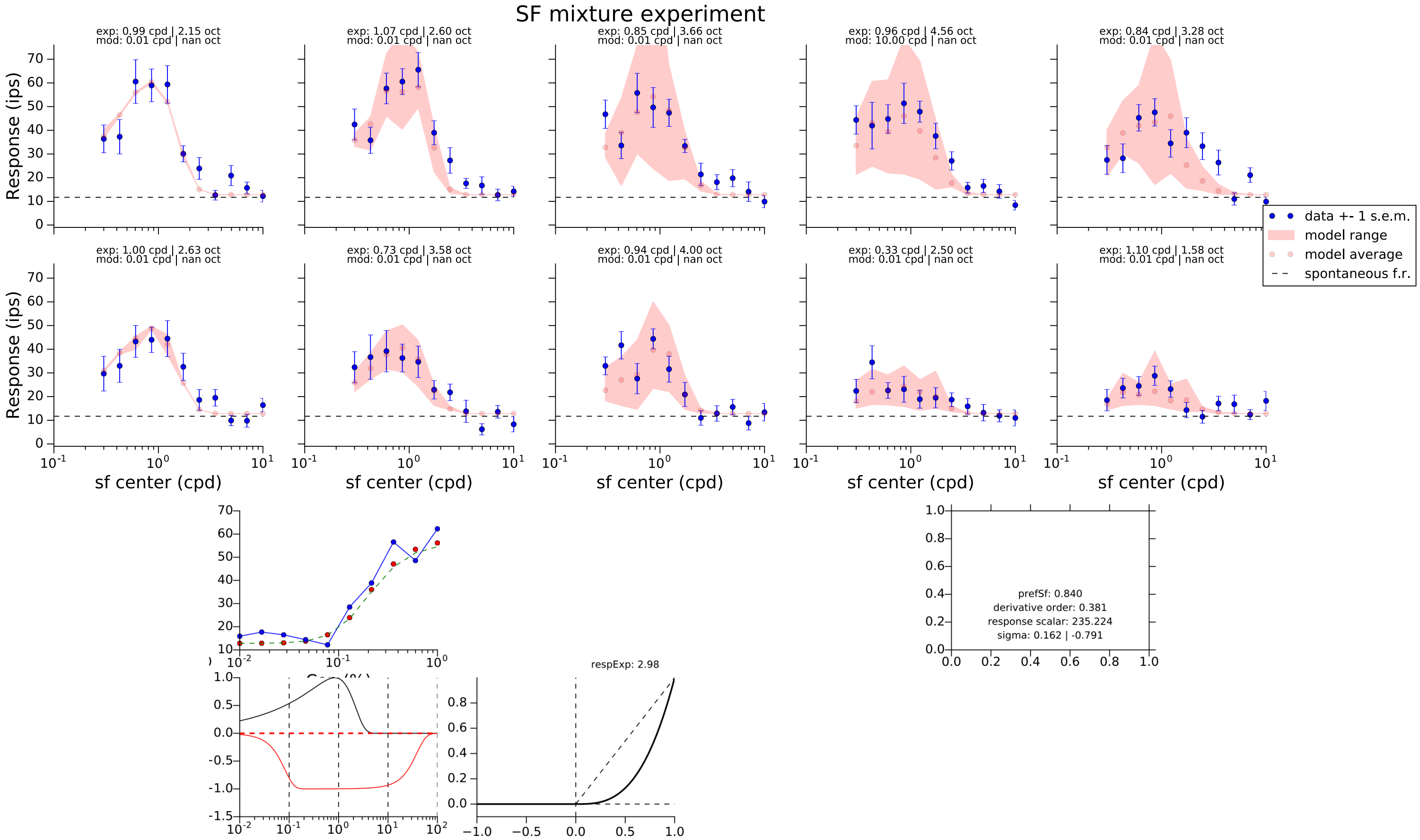


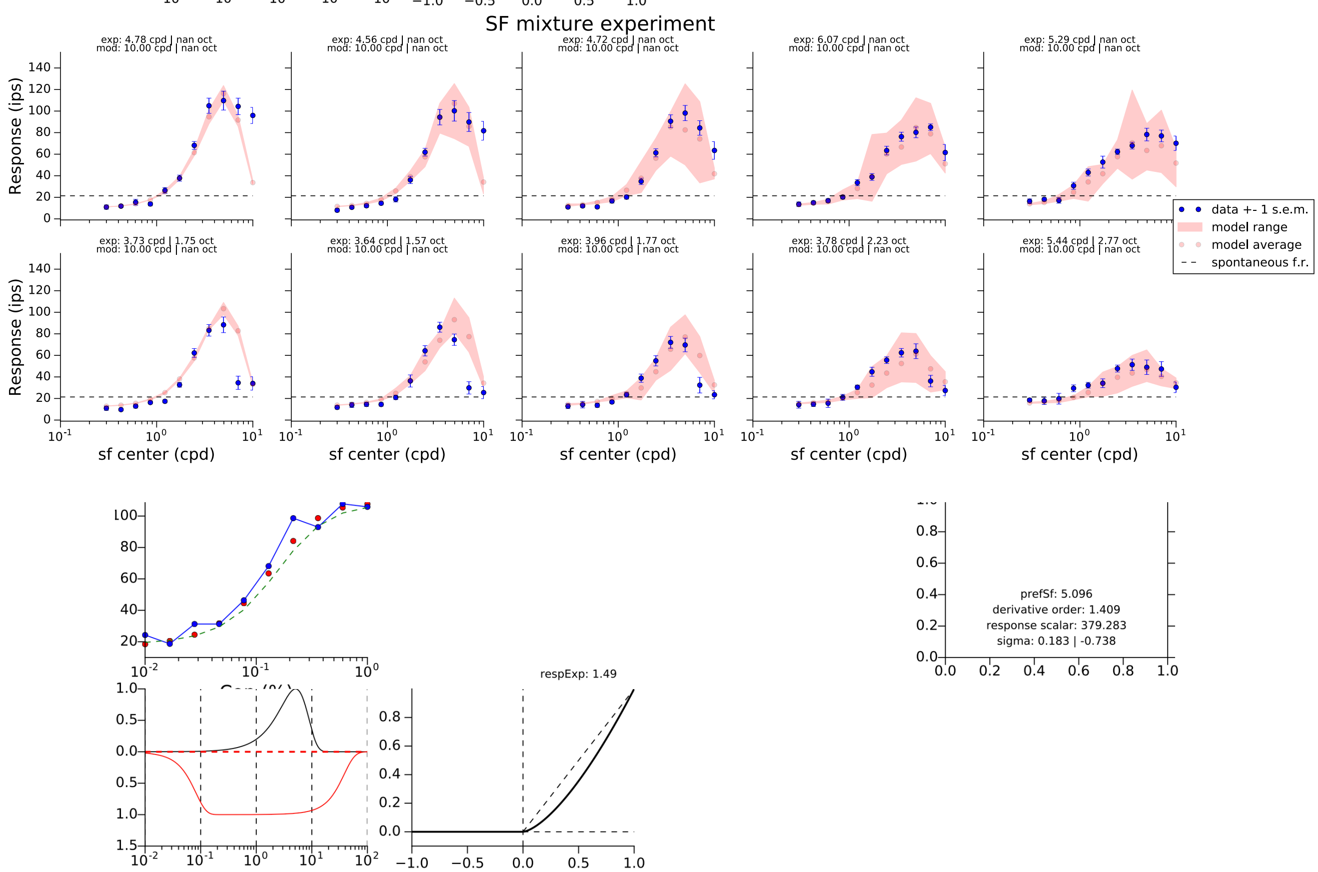


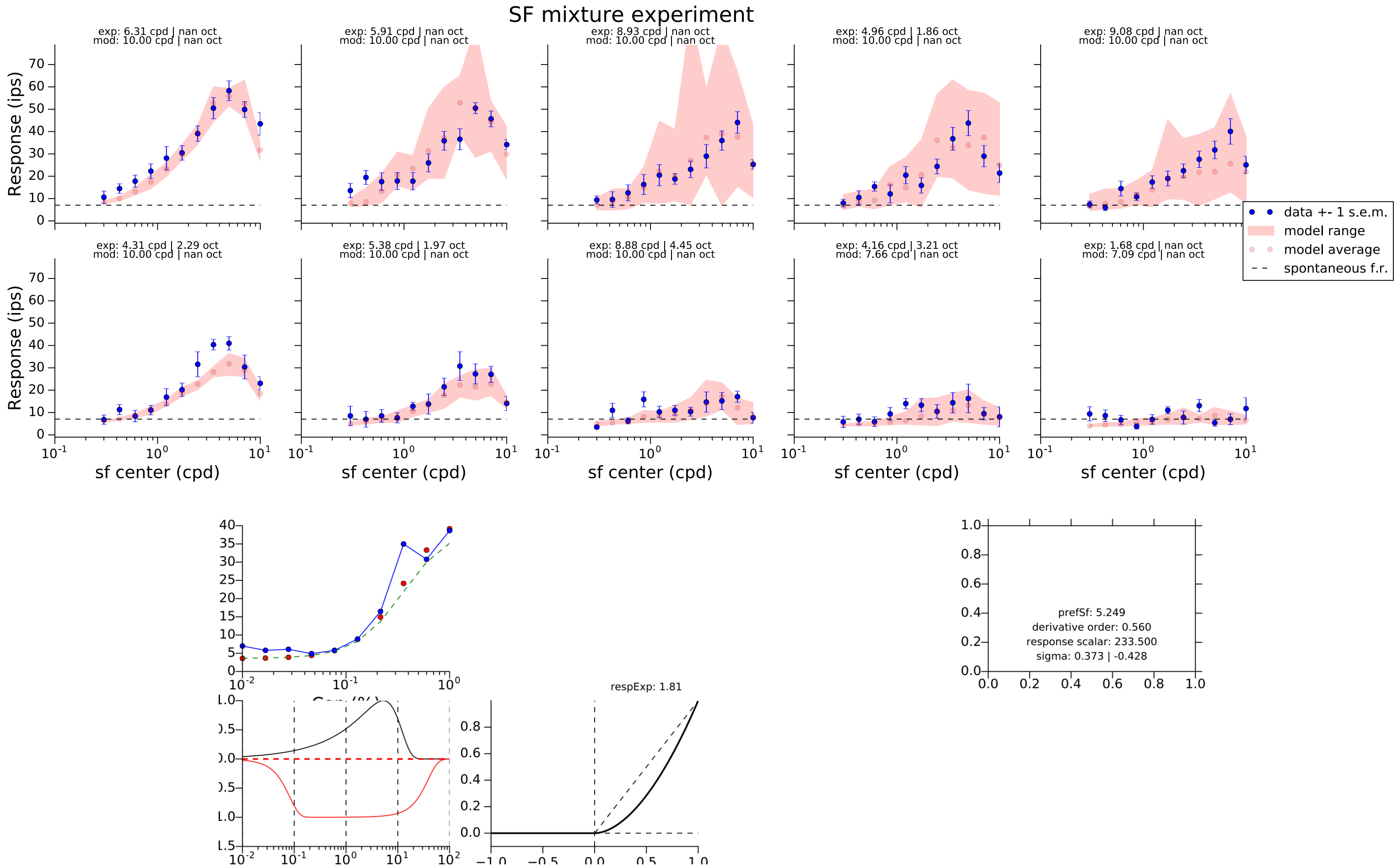

SF mixture experiment
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
exp: 6.31 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.91 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 8.93 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.96 cpd | 1.86 oct
mod: 10.00 cpd | nan oct
exp: 9.08 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.31 cpd | 2.29 oct
mod: 10.00 cpd | nan oct
exp: 5.38 cpd | 1.97 oct
mod: 10.00 cpd | nan oct
exp: 8.88 cpd | 4.45 oct
mod: 10.00 cpd | nan oct
exp: 4.16 cpd | 3.21 oct
mod: 7.66 cpd | nan oct
exp: 1.68 cpd | nan oct
mod: 7.09 cpd | nan oct
Response (ips)
sf center (cpd)
prefSf: 5.269
derivative order: 0.562
response scalar: 255.145
sigma: 0.357 | -0.448
respExp: 2.00

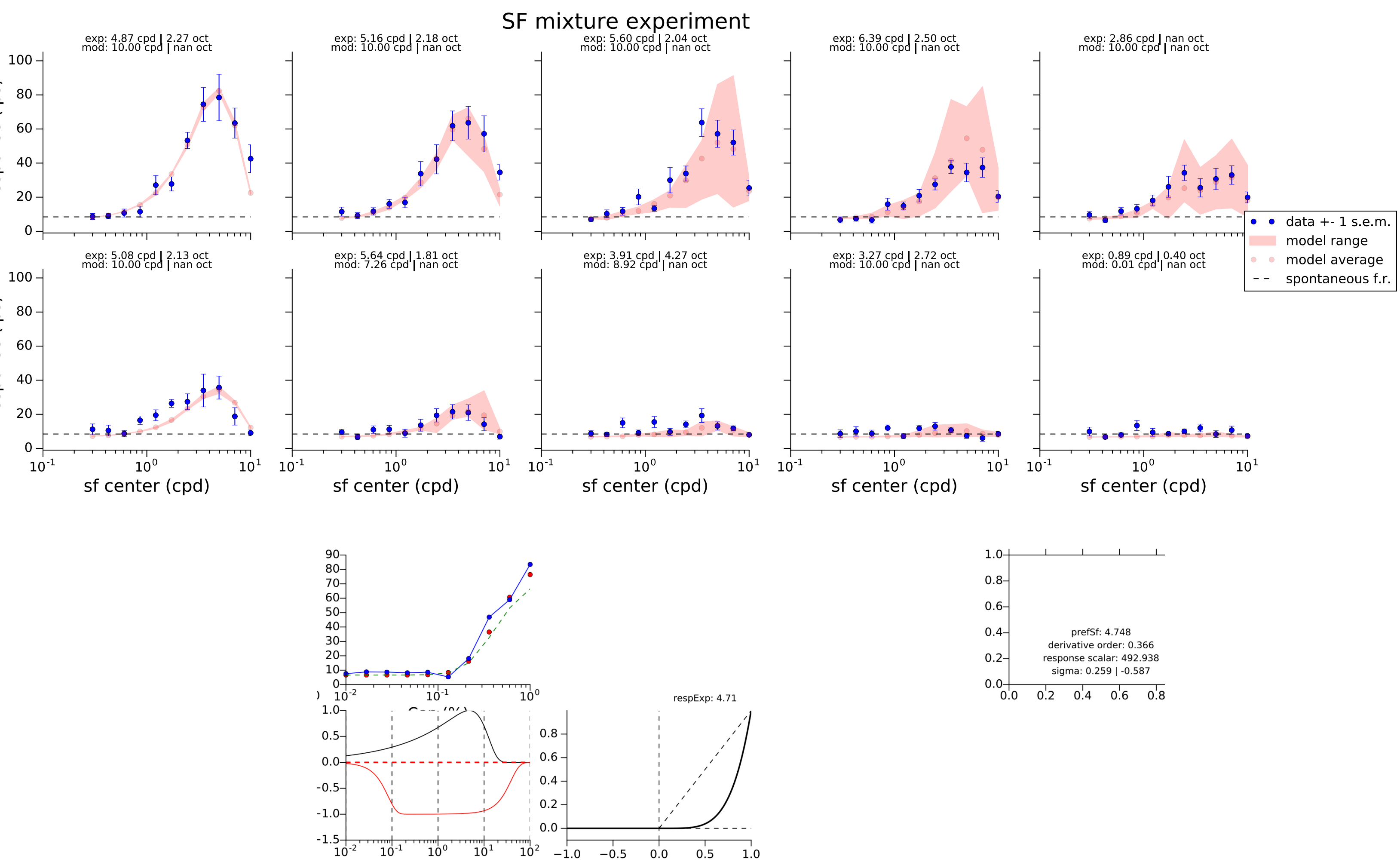

SF mixture experiment
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
exp: 4.87 cpd | 2.27 oct
mod: 10.00 cpd | nan oct
exp: 5.16 cpd | 2.18 oct
mod: 10.00 cpd | nan oct
exp: 5.60 cpd | 2.04 oct
mod: 10.00 cpd | nan oct
exp: 6.39 cpd | 2.50 oct
mod: 10.00 cpd | nan oct
exp: 2.86 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.08 cpd | 2.13 oct
mod: 10.00 cpd | nan oct
exp: 5.64 cpd | 1.81 oct
mod: 7.26 cpd | nan oct
exp: 3.91 cpd | 4.27 oct
mod: 8.92 cpd | nan oct
exp: 3.27 cpd | 2.72 oct
mod: 10.00 cpd | nan oct
exp: 0.89 cpd | 0.40 oct
mod: 0.01 cpd | nan oct
Response (ips)
sf center (cpd)
prefSf: 4.886
derivative order: 0.806
response scalar: 422.293
sigma: 0.529 | -0.277
respExp: 2.00

Overweight the contrast response function (x100)

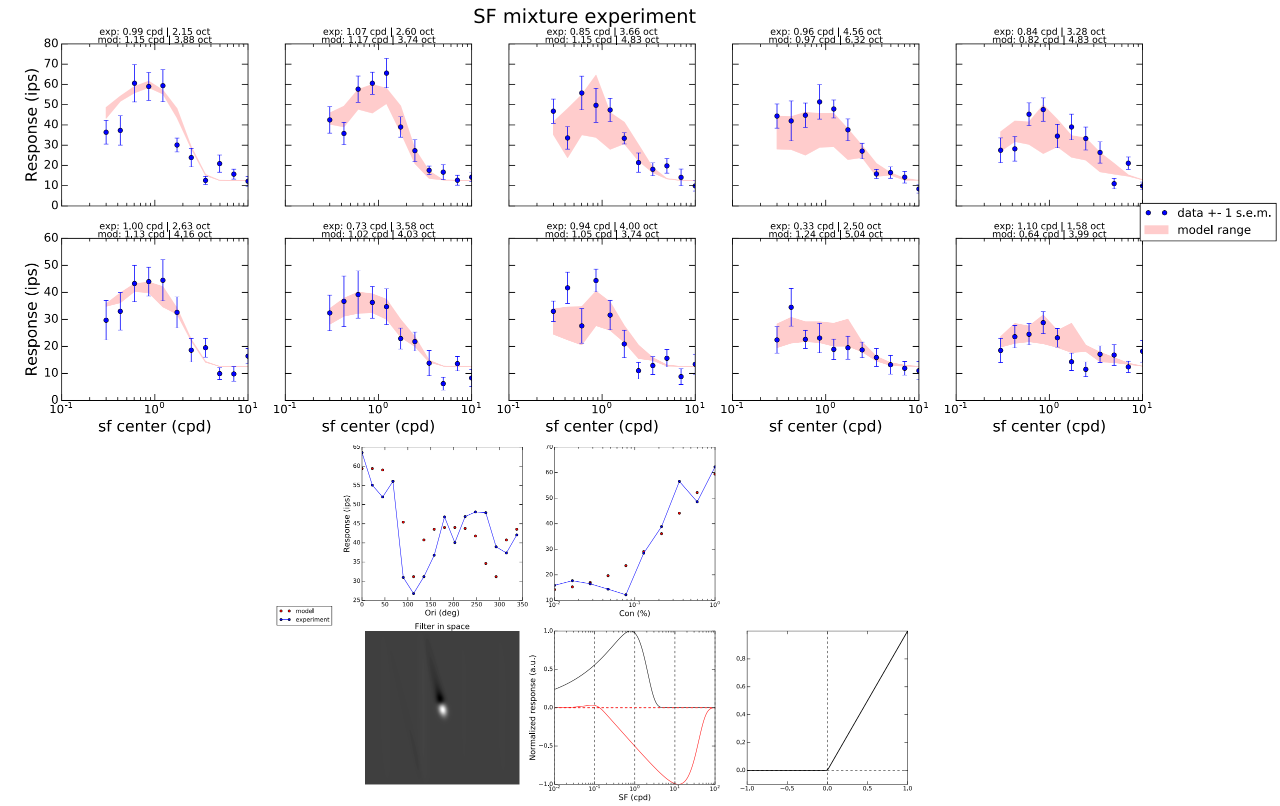


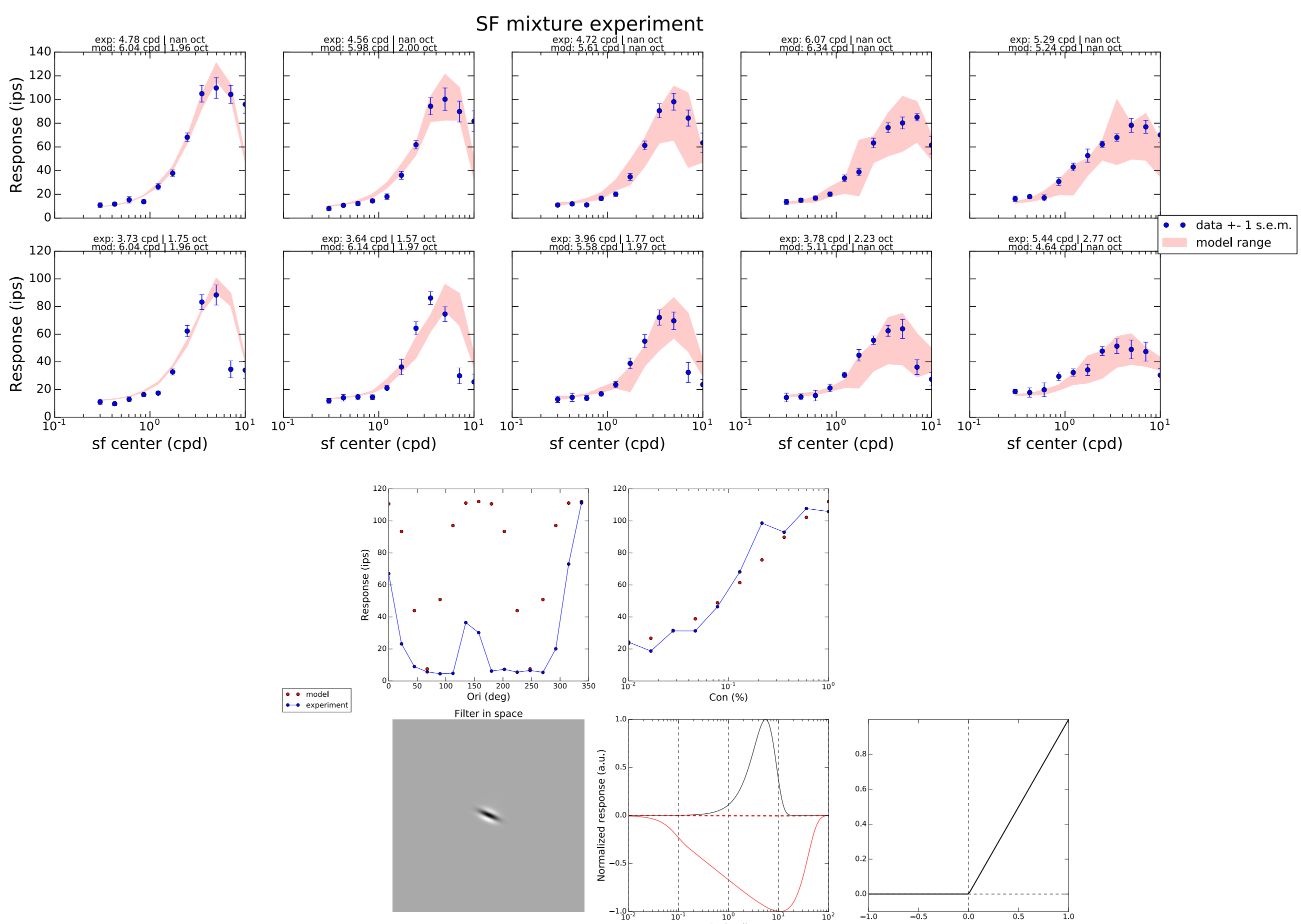


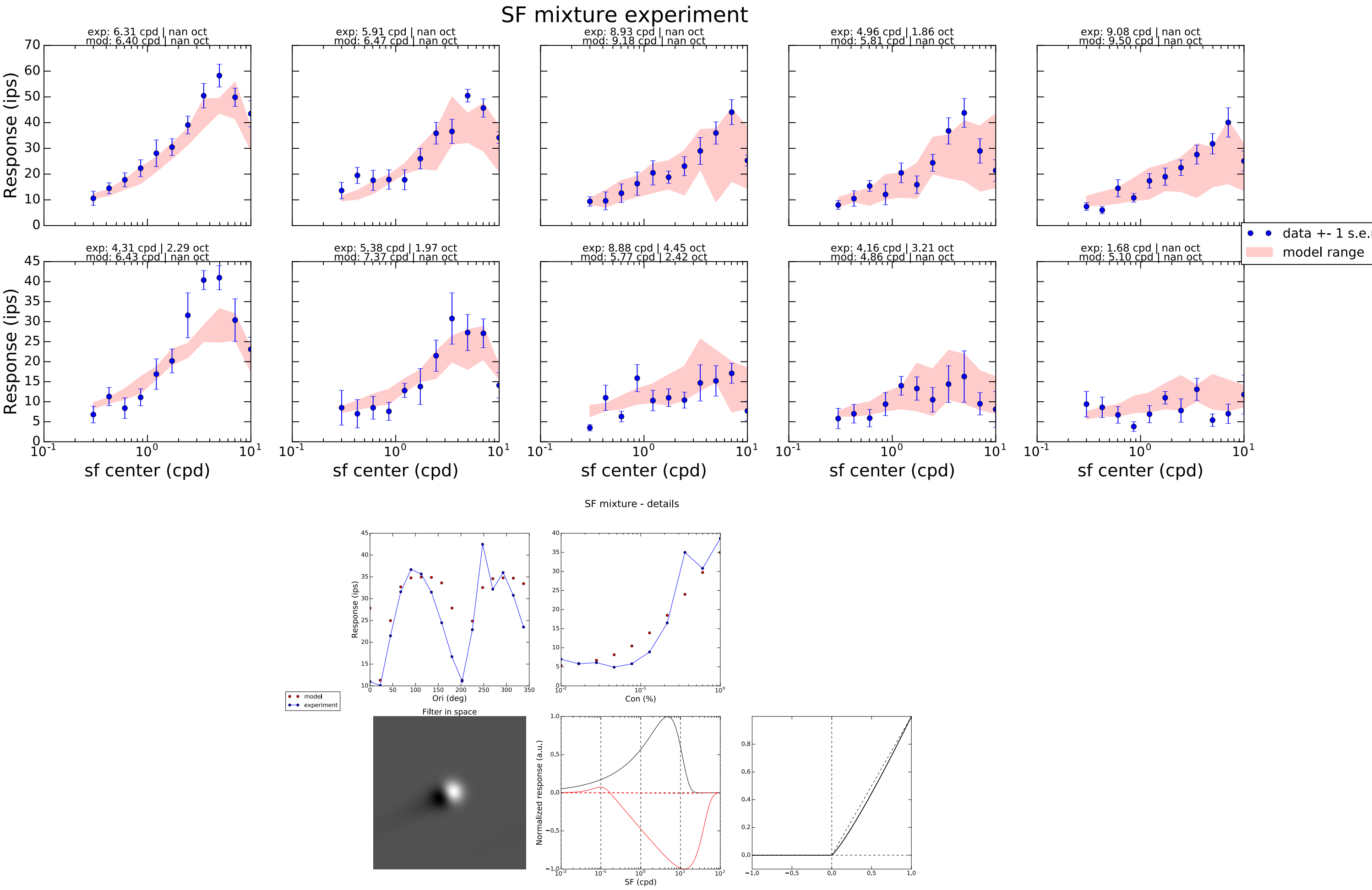

SF mixture experiment
exp: 6.31 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.91 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 8.93 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.96 cpd | 1.86 oct
mod: 10.00 cpd | nan oct
exp: 9.08 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 4.31 cpd | 2.29 oct
mod: 10.00 cpd | nan oct
exp: 5.38 cpd | 1.97 oct
mod: 10.00 cpd | nan oct
exp: 8.88 cpd | 4.45 oct
mod: 10.00 cpd | nan oct
exp: 4.16 cpd | 3.21 oct
mod: 7.66 cpd | nan oct
exp: 1.68 cpd | nan oct
mod: 7.09 cpd | nan oct
Response (ips)
sf center (cpd)
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
prefSf: 5.249
derivative order: 0.560
response scalar: 233.500
sigma: 0.373 | -0.428
respExp: 1.81

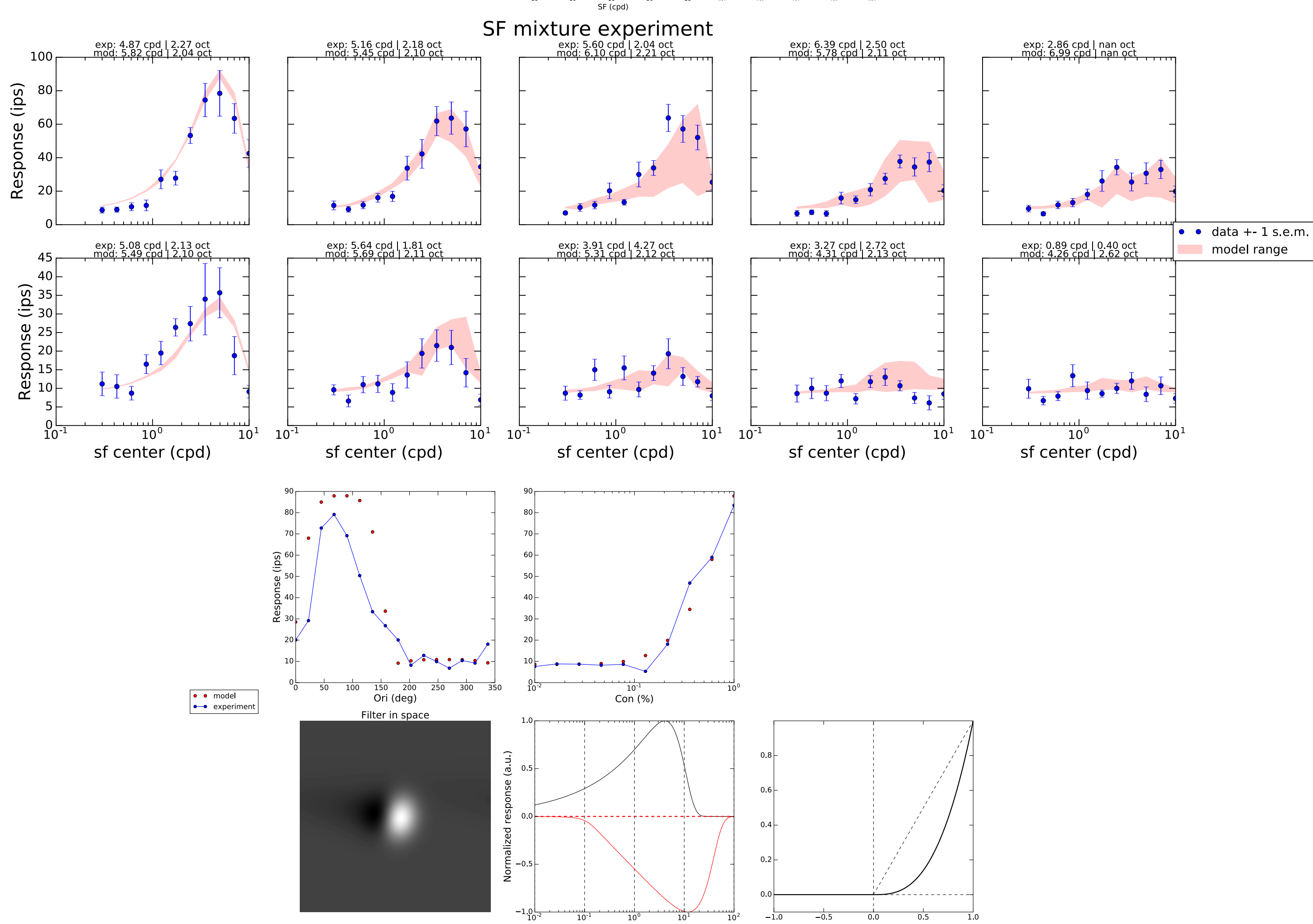

SF mixture experiment
exp: 4.87 cpd | 2.27 oct
mod: 10.00 cpd | nan oct
exp: 5.16 cpd | 2.18 oct
mod: 10.00 cpd | nan oct
exp: 5.60 cpd | 2.04 oct
mod: 10.00 cpd | nan oct
exp: 6.39 cpd | 2.50 oct
mod: 10.00 cpd | nan oct
exp: 2.86 cpd | nan oct
mod: 10.00 cpd | nan oct
exp: 5.08 cpd | 2.13 oct
mod: 10.00 cpd | nan oct
exp: 5.64 cpd | 1.81 oct
mod: 7.26 cpd | nan oct
exp: 3.91 cpd | 4.27 oct
mod: 8.92 cpd | nan oct
exp: 3.27 cpd | 2.72 oct
mod: 10.00 cpd | nan oct
exp: 0.89 cpd | 0.40 oct
mod: 0.01 cpd | nan oct
sf center (cpd)
data +- 1 s.e.m.
model range
model average
spontaneous f.r.
prefSf: 4.748
derivative order: 0.366
response scalar: 492.938
sigma: 0.259 | -0.587
respExp: 4.71

Original model - normalization, subtractive suppression, orientation fits






SF mixture experiment
exp: 6.31 cpd | nan oct
mod: 6.40 cpd | nan oct
exp: 5.91 cpd | nan oct
mod: 6.47 cpd | nan oct
exp: 8.93 cpd | nan oct
mod: 9.18 cpd | nan oct
exp: 4.96 cpd | 1.86 oct
mod: 5.81 cpd | nan oct
exp: 9.08 cpd | nan oct
mod: 9.50 cpd | nan oct
exp: 4.31 cpd | 2.29 oct
mod: 6.43 cpd | nan oct
exp: 5.38 cpd | 1.97 oct
mod: 7.37 cpd | nan oct
exp: 8.88 cpd | 4.45 oct
mod: 5.77 cpd | 2.42 oct
exp: 4.16 cpd | 3.21 oct
mod: 4.86 cpd | nan oct
exp: 1.68 cpd | nan oct
mod: 5.10 cpd | nan oct
Response (ips)
sf center (cpd)
model range
SF mixture - details
Response (ips)
Ori (deg)
Con (%)
model
experiment
Filter in space
Normalized response (a.u.)
SF (cpd)


SF mixture experiment
exp: 4.87 cpd | 2.27 oct
mod: 5.82 cpd | 2.04 oct
exp: 5.16 cpd | 2.18 oct
mod: 5.45 cpd | 2.10 oct
exp: 5.60 cpd | 2.04 oct
mod: 6.10 cpd | 2.21 oct
exp: 6.39 cpd | 2.50 oct
mod: 5.78 cpd | 2.11 oct
exp: 2.86 cpd | nan oct
mod: 6.99 cpd | nan oct
exp: 5.08 cpd | 2.13 oct
mod: 5.49 cpd | 2.10 oct
exp: 5.64 cpd | 1.81 oct
mod: 5.69 cpd | 2.11 oct
exp: 3.91 cpd | 4.27 oct
mod: 5.31 cpd | 2.12 oct
exp: 3.27 cpd | 2.72 oct
mod: 4.31 cpd | 2.13 oct
exp: 0.89 cpd | 0.40 oct
mod: 4.26 cpd | 2.62 oct
Response (ips)
sf center (cpd)
data +- 1 s.e.m.
model range
Response (ips)
Ori (deg)
Con (%)
model
experiment
Filter in space
Normalized response (a.u.)
SF (cpd)

Summary and future

I have summarized the basic motivation, history, and current status of our efforts to model spatial frequency tuning in macaque V1. The above sets (5) of model fits demonstrate a subset of the attempted model configurations and their effects on 4 example cells. Though other cells may have exhibited different behavior in the different configurations - for example, some cells had very poor, non-converging or local-minima-stuck fits - I tried to pick cells which were representative.

The four example cells exhibit a range of tuning properties from low to high pass, narrow to broad, steep contrast gain to shallow, etc. The model in its current form does reasonably well in capturing the range of behaviors. For example, both cells 47 and 57 are reasonably selective for spatial frequency, but only the latter has a steep response nonlinearity, likely a means of capturing the steeper drop in responses observed for this cell with changes in dispersion and contrast. The model's contrast response functions are another key success, at least for these cells. Nonetheless, there are failures. For example, note the inability of the model to capture the sustained/present response at the high frequency end of the curve for cells 27 and 57. Additionally, note the failure of the model to capture changes in response amplitude and tuning shape at low contrast for cells 47 and 57.

Though introducing an additional parameter should in principle never make a fit worse, it appears that the freedom of asymmetric gain over spatial frequency for the normalization pool does exactly this. Granted, the fits were started fresh (rather than based off of the better fits without this parameter) and an additional optimization with a finer stopping criterion was not run in this configuration, but the model seems to find bad local minima. In all four cells, the asymmetry parameter has reached a bound and generally the fits - in shape and response amplitude - are significantly worse here.

One way of exploring the model and its fitting procedure is to fix or eliminate parameters to see where it can "trade-off". We attempted this by fixing the response nonlinearity to 2 and optimizing all other parameters. Noting that cells 27 and 47 have a response exponent of 1 in the current fits, when fixing this to 2, the trade-off is a reduction in derivative order. Regardless, the fits do not suffer greatly in any of these 4 cells, perhaps indicating the insensitivity of the model to this parameter or the ability of this parameter to freely - and seemingly penalty-free - trade the effect(s) of this parameter with other parameters, at least for these cells.

Attempting to "overfit" the contrast response function in order to better capture changes in response amplitude were not successful. Though the fits in these cells aren't worse, by any stretch, the improvement in contrast response fit also wasn't substantial. Without overweighting, the fits to the contrast response function are already successful, and this steeper weighting could result in devaluing other important features of the neural responses.

The full, original model with fits to the orientation data, asymmetric normalization, and subtractive suppression, were reasonably "good" for these four example cells, though it's worth pointing out that these are among the better cases; it was more common in this model configuration for the fits to be worse than they are in the current configuration. The fit of cell 47 is instructive, though: the model poorly captures changes in response amplitude with changes in contrast and dispersion. To make matters worse, this is accomplished with more parameters than in our current, simpler iteration.

Going forward, we can take insights from these various exercises. I have tried other configurations not described here - for example, subtracting a constant term (rather than a response which looks like the normalization signal) - and found that our current, simple model is a very worthwhile starting point. With the fits largely behaving, though far from perfect, we can start to reintroduce model complexity or make adjustments to fix the existing issues. For example, though asymmetric normalization does not work well, perhaps a normalization signal which is peaked at or near the filter's peak could help capture amplitude changes (e.g. Schwartz and Simoncelli, 2001). We can also likely reintroduce optimization for the orientation tuning data so that we can get a richer picture of the full 2D kernel (i.e. preferred orientation, direction selectivity, aspect ratio). The seeming robustness or degeneracy of the model (which is it?) to certain manipulations like fixing the response exponent are worth exploring; I don't believe the model needs to be re-parameterized, but it is worth discussing.

The larger "elephant in the room" is the inability of the model to capture shifts in preferred spatial frequency with contrast. I believe that a gain control mechanism like the one described in Schwartz and Simoncelli might help with this (right?) though other mechanisms also serve as candidates. For example, perhaps as a start, we can build in a parameter which allows a contrast dependent shift in the peak of the filter. Assuming this proof-of-concept works, we can then try to introduce mechanistic plausibility - perhaps a Sach Sokol-like parameterization of some LGN neurons can serve as a template for contrast dependent shifts. Such a model will certainly be more difficult, but if properly grounded, it can serve as a useful tool for gaining intuition to guide our planned LGN recordings and modeling.