**Widefield Sign Mapping User Guide**

Written 31 Aug 2017 KS

Updated:

**Purpose**

Complete MATLAB based analysis of sweeping bars across a monitor in the four cardinal directions in order to extract a visual field sign map, allowing us to segment and accurately place visual cortices on a mouse-by-mouse basis.

**Included Functions**

*signMapMaster*

A master file containing calls to all the below functions in the correct order and method of running them. Each member of the processing pipeline can also be separately run at any step, to process data from a specific step onward. Also has a single changeable parameter: fs, the acquisition framerate

*widefieldDFF*

Reads in your multi-page tif file, then calculates a median-based average activity map of your window (F0). Then a frame by frame DFF matrix is constructed via the standard DFF function:

Where Ft is the fluorescence at a given time, t (in this case, an entire frame of fluorescence), and F0 is the average fluorescence over the entire recording session, calculated above.

*widefieldRespSeparator*

Given the information from the stimulus data file, this function separates your data into the different response directions, separating horizontal left to right, right to left and vertical top to bottom, and bottom to top. Of each axis (horizontal and vertical) one of the response directions is flipped along the dimension of frames in order to allow meaning of the forward and backward directions. This further allows reduction of the positive skew that is typical of DFF data. The data are then baseline subtracted from the mean of the values in the preceding 2s off-time on a repetition by repetition basis. Lastly, all the horizontal responses and vertical responses are separately concatenated, and the mean across repetitions is performed to give us the data for Fourier transforming.

*signMapCombiner*

The above three functions can be run many times on as many blocks that you have (signMapMaster will prompt you to choose each set of data). Afterwards, signMapCombiner will combine the response data from each block and mean them to get an average response across blocks to use for Fourier transform.

*A\_widefieldFourierTransform*

The first step of the main sign mapping pipeline, A\_widefieldFourierTransform quickly conducts a Fourier transform in the frames dimension on each of the data separately, and extracts the correctly scaled magnitude and phase informations from the Fourier transform.

*B\_widefieldPhaseMap*

Because the Fourier transformed data contain many frequency bins which are present in the signal (including noise), the next step is manual curation and selection of the correct frequency bin for further processing. This is achieved via a GUI which will display different phase maps at certain frequency bins to allow you to choose one. Along with this, the represented actual frequency will be described at the top to help you choose the correct phase map. Almost always, it is the second frequency bin which is the correct bin (which corresponds to the first Fourier transformed harmonic). One at a time, the GUI will ask you to choose a phase map for the azimuth (horizontal) phase map, and then the altitude (vertical) phase map. The frequency bin for these two should be the same. It will then display the two phase maps and ask you for confirmation that these are the correct maps going forward.

*C\_widefieldSignMap*

The final step in the pipeline, C\_widefieldSignMap constructs sign maps based on your chosen phase maps above. The gradient in each of the primary axes is taken, then the angle between the gradients for each map is found. The sine of the difference between the angle for the azimuth vs the altitude map gives us the visual field sign map. Some filtering is done, then the map is displayed for you to examine. The map is then processed further via normalization, binarization, and cleaning up of stray pixels. Lastly, the borders of your sign map is then plotted on top of a reference image chosen earlier to give a precise idea of where the visual cortices lie on your imaging plane.

**Inputs**

*Multi-page .tif image file*

Image file from the recording session. One tif per frame.

*Stimulus data file*

A stimulus data file which is produced by the stimulus file, contains necessary information for the analysis of data.

*Reference surface veins image*

A screenshot or other picture (preferably square) of the surface veins of the imaging plane. If not provided, the code will provide a reference image using the F0 map, but this will be defocused from the surface plane, and provide less resolution in identifying areas for future use.

**Step by step guide**

1. Presentation of stimulus and acquisition of multi-page tif and stimulus data file. (For the included sample data, there will be an example\_images.tif file, example\_stimdat.mat file, and an example\_refimg.jpg file for you to use).
2. Run signMapMaster. It will prompt you to ask how many recording blocks you’ve done (you can input 1 if you just did 1), then ask you to choose the corresponding multi-page tif for each block. Although you can have a stimulus data file for each block of the stimulus, the data in each stimulus data file must be identical, or you cannot collect results across blocks; therefore, the code only prompts once for the stimulus data file. Lastly, it will prompt you to choose a reference image file, as mentioned above, if you forgot or did not take a reference image file, just answer “No”, and the F0 map will be used in lieu of a reference image.
3. **Step 0: Calculating DFF matrix and Extracting Data** – This part will run N times, where N is the number of blocks you’ve recorded. It will calculate and save the response data in each direction (azimuth and altitude) for each block for further use. As mentioned above, if your acquisition framerate is NOT 10Hz, ensure that you change the value of fs in signMapMaster. Lastly, Step 0 will combine all your recording blocks into a single matrix for further processing
4. **Step A: Fourier transforming the data** – This step is very fast. See above for description of the process, but you will not have to provide any input.
5. **Step B: Phase map selection and** **creation** – The code will automatically select the phase map of the second frequency bin, which should correspond to your stimulus frequency. It will display these to you and ask if the phase maps look good. See figure 1 for an example of a good phase map. If the phase maps look wrong, it’s because there was another signal with a higher magnitude (noise or something else) that drowned out your stimulus signal and we then have to enter manual curation. Upon choosing “No”, it will launch the manual curation GUI. The only manual step in the pipeline, the GUI will prompt you to choose your phase map of interest. (Figure 2) Use the Previous and Next buttons to change your frequency bin (shown at the top by the k-value). The corresponding real frequency for this recording would be ~ 0.083 Hz (1/12s). See Figure 3 for an example of how an azimuth phase map should roughly look. More important is the correct frequency displayed at the top. As mentioned above, the correct frequency should almost always be corresponding to k = 2. After repeating this step for your altitude phase map as well, you will be presented with your two maps side by side (Figure 4). Press finished to complete Step B, or press azimuth or altitude to return to one or the other to re-choose the map. Your chosen phase maps will be saved as a JPG.
6. **Step C: Creating the sign map and post-processing** – Your phase maps will then be automatically taken and the sign map will be calculated (Figure 5) and saved as a JPG. Press any button to continue, and the post-processing will happen and present you with a border overlaid on your reference image. This will also be saved as a JPG. This concludes the processing pipeline.
7. **Manual post-processing –** A useful post processing step is to use a reference sign map from a more established paper (see Zhuang et al 2017, Garrett et al 2014 for examples) to label different visual areas in your own sign map. Most simple is to use the visual field sign map (VFS) as opposed to the overlay map. Note the relative size and position of the visual areas to each other, as well as the sign of each area. For example, V1 is characteristically negative (blue) in sign, this will always be true. Visual areas retain their sign over recordings and across mice. This will help in determining which areas are separate and which are together. (See Figure 6)

**Contact**

For any troubleshooting or help with the stimulus presentation, you can contact Raquel Abreu at desousaabreu@lifesci.ucsb.edu

For any troubleshooting or help with the analysis code or technical help, you can contact Kevin Sit at kevin.sit@lifesci.ucsb.edu

**References**

Zhuang, J., Ng, L., Williams, D., Valley, M., Li, Y., Garrett, M., & Waters, J. (2017). An extended retinotopic map of mouse cortex. *eLife*, *6*, 1–29. https://doi.org/10.7554/eLife.18372

Garrett, M. E., Nauhaus, I., Marshel, J. H., & Callaway, E. M. (2014). Topography and areal organization of mouse visual cortex. *Journal of Neuroscience*, *34*(37), 12587–12600. https://doi.org/10.1523/JNEUROSCI.1124-14.2014

**Figures**

**Figure 1**

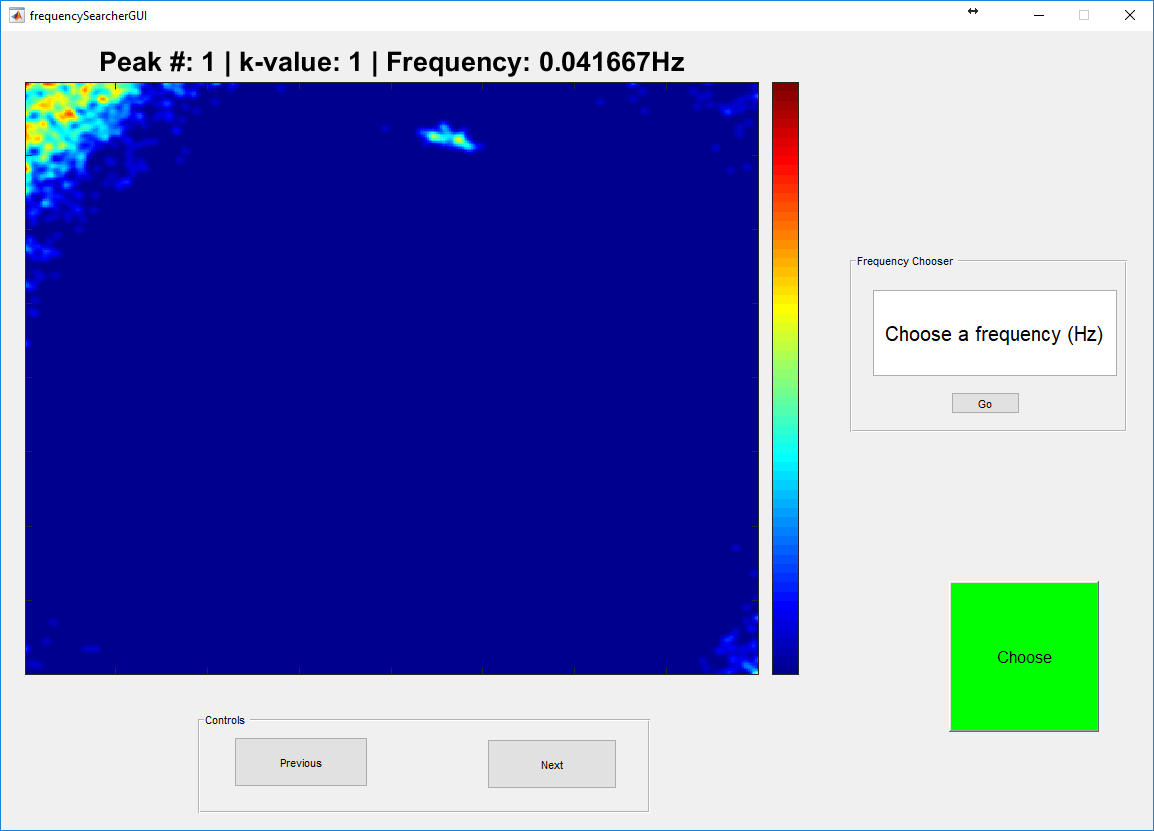


Figure 2: GUI for choosing the phase map of interest. Previous and Next allow you to go back and forth through the frequency bins to choose the correct phase map of interest. If necessary, you can type in your expected frequency of interest, and the GUI will display the phase map closest in frequency to your inputted frequency.

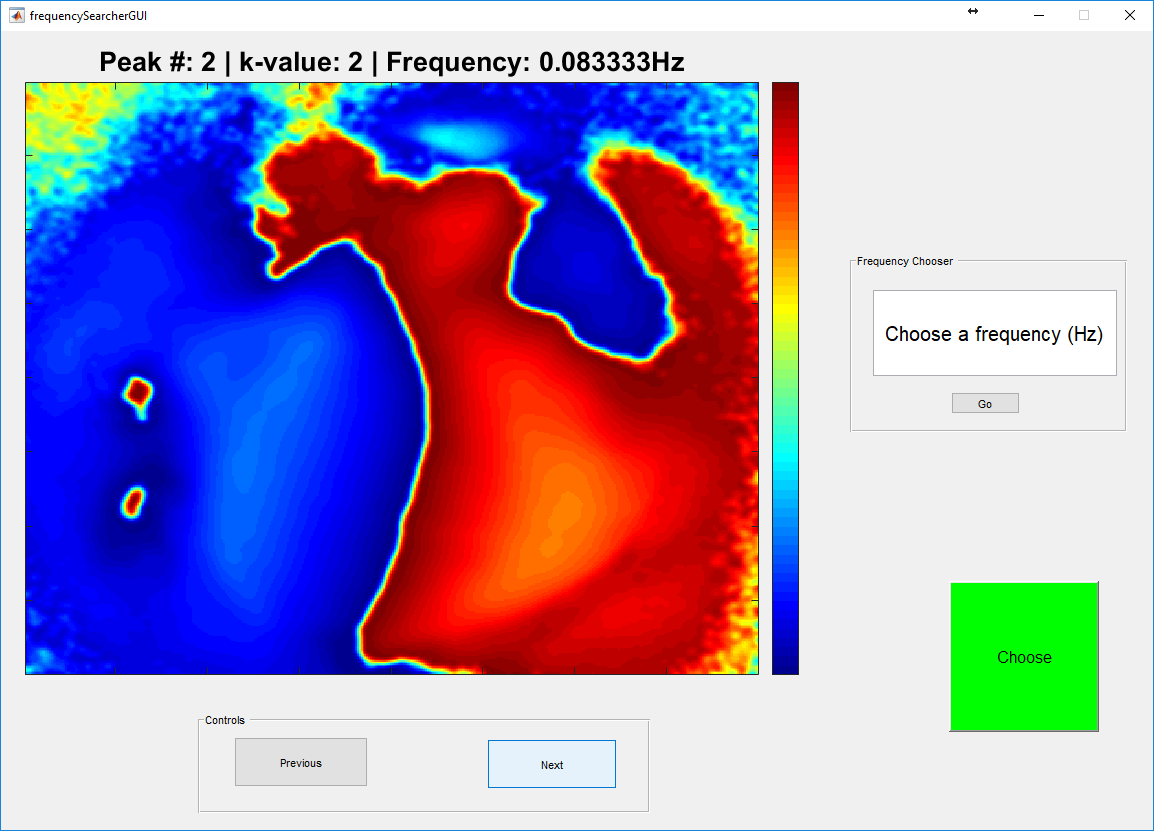


Figure 3: An example of a good phase map for the azimuth direction. Note the clean separation between the nasal and temporal areas of the cortices. Phase maps should show clean separation of the visual cortices into two discrete sections, as shown above. Because this is a good map, pressing the green “Choose” button will select this frequency bin as the correct frequency bin of interest. Remember, the frequency bin you choose for one axis should be the same as for the other axis.

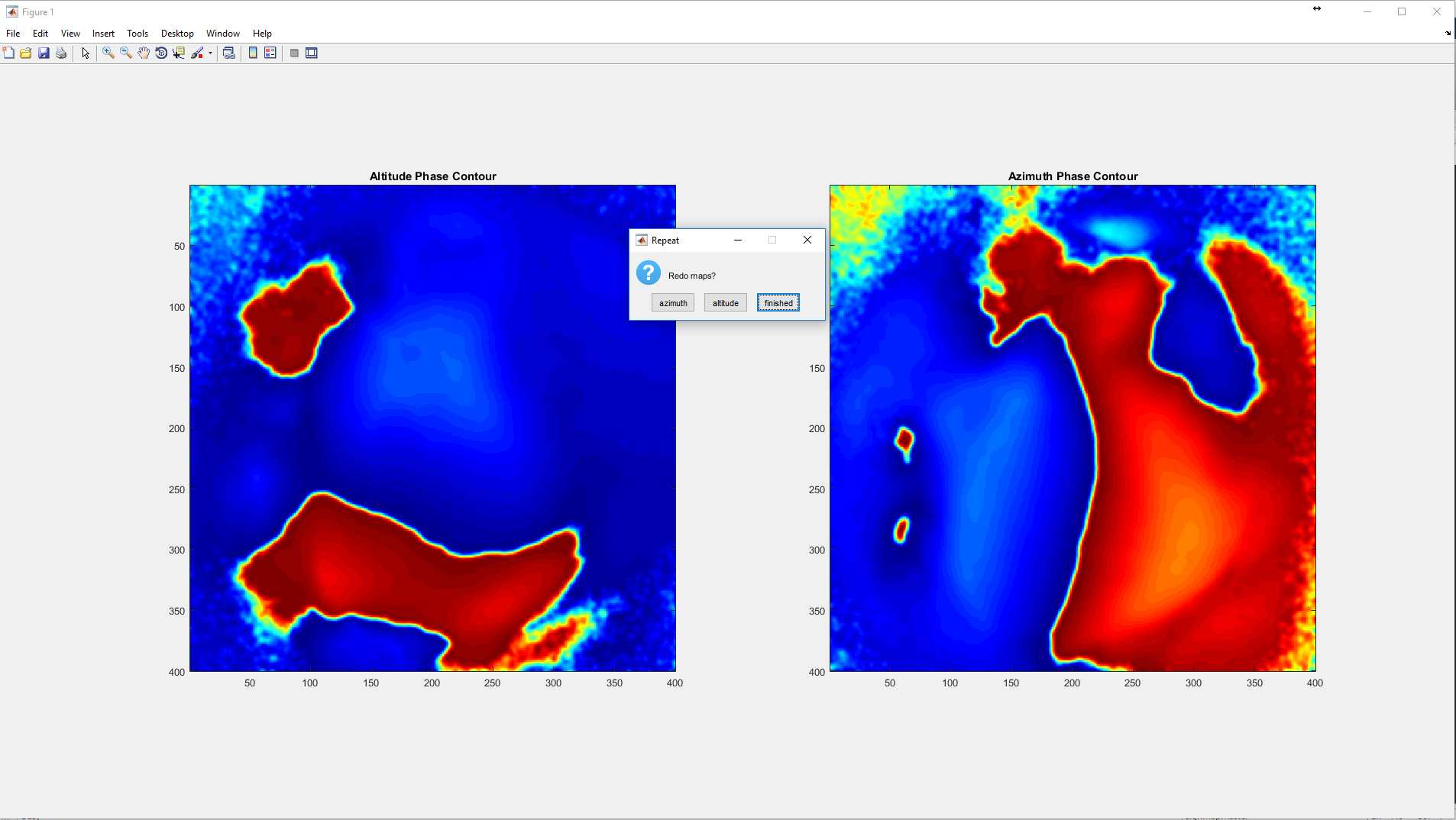


Figure 4: The two phase maps displayed side by side. Most importantly, ensure that the areas which the two sign maps encompass seem reasonably overlapping, indicated they are both from visual cortex. Rarely, you will have to return to the GUI to re-choose the maps.

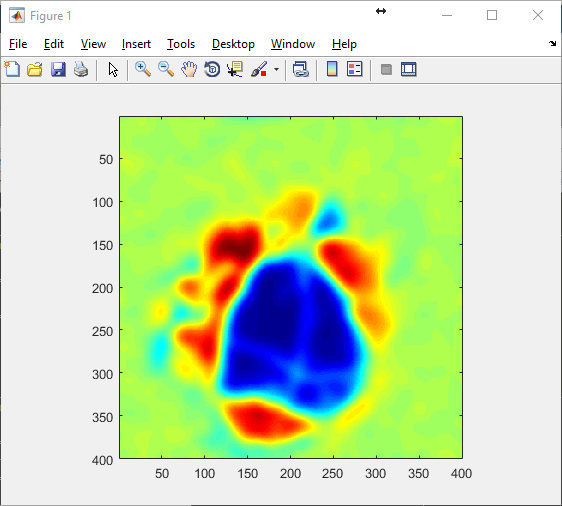


Figure 5: An example sign map which clearly shows several cortical areas. Note also the lack of any meaningful sign values outside of visual cortices.

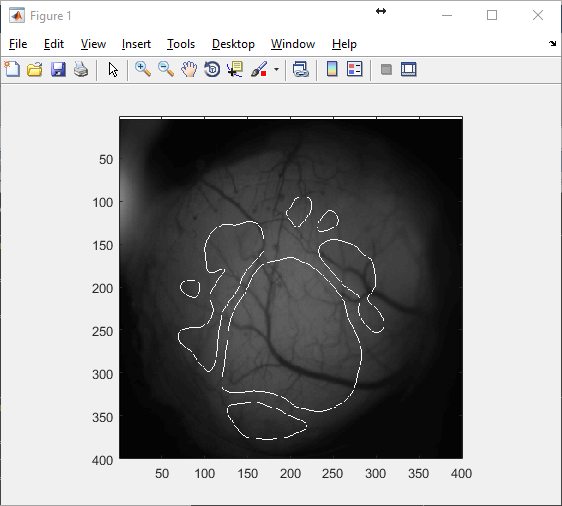


Figure 6: An example of the overlay on the reference image. This gives a rough estimate of the location of different visual areas and allows more directed imaging in the future.

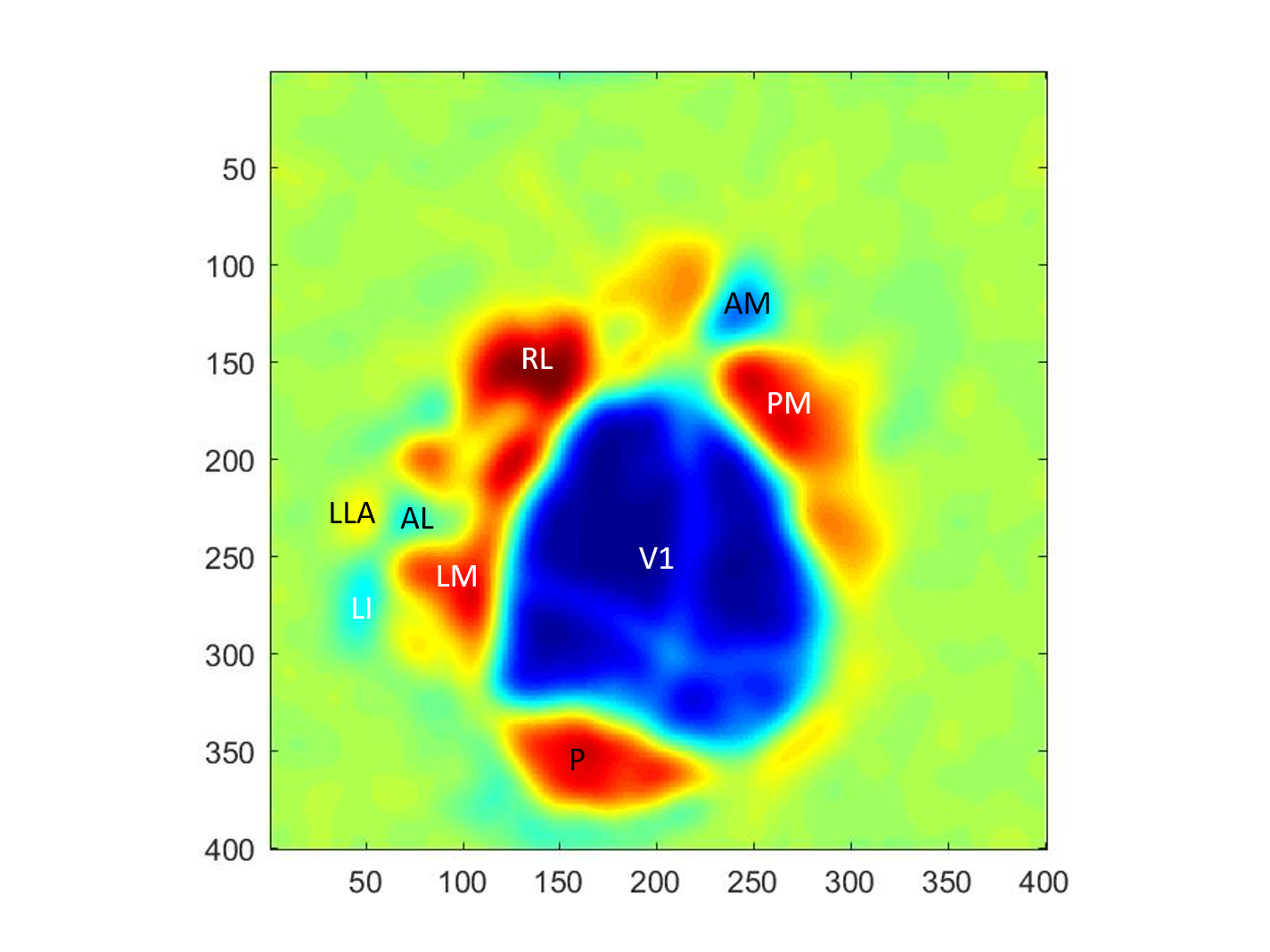


Figure 7: A manually labeled sign map (using Zhuang et al 2017 as reference) that gives a good rough estimate of the locations of different visual areas.