Week 8 Exercises

This week's exercises will investigate the effects of signal averaging. You will work with a real dataset collected in a simultaneous visual/motor task. The exercises will require you to find active voxels in a dataset and evaluate the effects of sampling on those data.

The MATLAB functions needed are toexcel, find, corrcoef, max, mean, std, squeeze, for.
Scripts needed are compile_epochs, sort_epochs, average_epochs.

Exercise 1: Analyzing the MR data

1.1 Properties of the Data

The experimental data are 96 trials from a combined visual and motor task. On each trial, the subject saw a series of four flashes of a counterphase checkerboard, each presented for 500ms. The subject squeezed both hands each flash. Thus, each trial began with 2s of visual stimulation and 2s of motor movements. The time between trials was jittered between 15-18s, to allow the hemodynamic response to return to baseline. The experimental paradigm is included in as checker_test.txt in the directory:
T:\BIAC\Class.01\Examples\Functional_Data\Flash_squeeze

1.2. Running the tstatprofile analysis: All Runs

Header, paradigm, and weights files for you to use are available in the directory:
T:\BIAC\Class.01\Examples\Functional_Data\Flash_squeeze

You will need to copy the header file to a directory for your group so that you can change the output directory. You can change the other files at your discretion. The most likely file for you to change will be the weights file, which is a rough approximation of the hemodynamic response, but can be improved. **You should run tstatprofile on the changed header file**.

1.3. Running the tstatprofile analysis: All Runs

Rerun the tstatprofile analysis on only three of the runs (choose randomly). You must change your outpath so that you do not overwrite the output created in step 1.2. **How does the activation change when only analyzing half of the data?**

1.4. Identifying high-signal and low-signal voxels for analysis

You should examine your data in showsrs2 to identify one or more voxels that seem to be highly active. You can use the baselineavg file as your base image and the bin01.t file as your overlay. Look for voxels that have a high significance value, and look at their time courses. **Choose two voxels, one from motor cortex and one from visual cortex, that are active both when you look at all of your data and when you look at just three runs.** Also, select voxels that are less consistently active. **Choose an additional two voxels that are active when you look at all of the data (at a low significance voxel), but are non-significant when averaging just three runs.**

Highly significant (both all runs and three runs)
Pick one voxel in motor cortex: X ____________ Y ____________ Z ____________
Pick one voxel in visual cortex: X ____________ Y ____________ Z ____________

Barely significant (all runs/not in three runs)
Pick one voxel in motor cortex: X ____________ Y ____________ Z ____________
Pick one voxel in visual cortex: X ____________ Y ____________ Z ____________

1.4. Measuring signal and noise for these voxels

The percent change in a voxel is determined by the change relative to baseline. To find out the baseline value for these voxels, you can load in the average baseline image.

```
baseline_image = readmr;  %(you will need to select the baselineAvg.img in your directory)
baseline=baseline_image(x,y,z)
```

To find out the change (in MR units), you should just look at the average epoch, either in showsrs or in the command window as described below.

```
average_epoch=readtsv;  %(you should select the Avg_1_V0001.img file)
change=max(squeeze(average_epoch(x,y,z,:)))
```

To determine the proportional signal change, divide the two quantities.

```
proportion_change=change/baseline
```

To determine the standard deviation across time points, you can load in the standard deviation image created by tstatprofile. Enter the values into the table below. Calculate the signal noise ratio (or effect size) by dividing the mean MR_change by the mean standard deviation.

```
stdev_image=readmr;        % (you should load the grandSd.img file)
stdev = stdev_image(x,y,z)
effect = change/stdev
```

High significance
Motor: Baseline _______ MR_Change _______ % Change ________ Stdev ________ Effect _______
Visual: Baseline _______ MR_Change _______ % Change ________ Stdev ________ Effect _______
Low significance
Motor: Baseline _______ MR_Change _______ % Change ________ Stdev ________ Effect _______
Visual: Baseline _______ MR_Change _______ % Change ________ Stdev ________ Effect _______

How high of SNR do you observe? Is this a large SNR value or a small one, compared to other sorts of experiments that we might conduct? For example, would we expect a smaller or larger value in a memory experiment?

1.5. Activity in veins vs. parenchyma

Before you go further, you should evaluate the signal change that you have measured against that typical for fMRI studies. Normally in simple visual/motor tasks we observe signal change of about 1-3%. Some voxels in this data set have signal change that is much larger than these values. These voxels are likely to be associated with draining veins, rather than the more localized parenchyma (capillary bed).

Are the voxels that you have chosen likely to be draining veins? If so, find non-vein voxels and write their coordinates in the space below. If not, can you find some example voxels (list them) with much larger signal change? How large is the vein change compared to normal?

Exercise 2: Extracting epochs

2.1. Creating the set of epochs from the data

The epochs can be extracted from the raw data using the script "compile_epochs". This script is written specifically for these data, but the concept can be applied to any event-related data set. Tstatprofile does something similar when doing event-related analyses. **Create variables for all 4 of your voxels from Exercise 1.**

```
addpath T:\BIAC\Class.01\Examples\Scripts
data_visual=compile_epochs(x,y,z);   %where x,y, and z represent your visual coordinates from above
data_motor=compile_epochs(x,y,z);   %where x,y, and z represent your motor coordinates from above
```

2.2. Looking at the mean time courses

The output of compile_epochs is a matrix of 96 trials by 19 time points. You can look at the mean time course using the mean, figure, and plot commands. Plot the data to make sure that the mean you extracted matches what you saw earlier in showsrs2.

```
figure,plot(squeeze(mean(data_visual,1)))
figure,plot(squeeze(mean(data_motor,1)))
```

2.3. Examining the standard deviation across the epoch

To examine the standard deviation across the epoch, use the std command.

```
std_epoch = std(data_visual);
```

Plot the standard deviation epoch. How does the standard deviation change across the epoch? Is the standard deviation larger in the baseline period or the task period? How much larger? Why?

2.4. Viewing the raw epochs

To quickly scroll through a number of plots, you can display each epoch one at a time within a loop. Use the return key to go to the next plot, and <ctrl>+C to exit the loop.

```
figure, for i = 1:96, plot(data_visual(i,:)),pause,end
```

Do individual trials look like the hemodynamic response that we have discussed already? What proportion (approximate) of trials have a somewhat recognizable hemodynamic response?

Exercise 3: Signal Averaging

This exercise will investigate how the ability to identify activation improves with signal averaging.

3.1. Using the sort_epochs script

The short script "sort_epochs" allows you to select randomly N rows from a row by column data set. So, you could use this script to select 10 trials from a set of 96. The script is in the "T:\BIAC\Class.01\Examples\Scripts" directory, so you will need to add that to the path if you have not already done so. **Examine a set of 10 trials and the mean of 10 trials from each of your voxels.**

```
output = sort_epochs(data_visual,10);
output_mean=mean(output,1);
figure, plot(output');   % Note that you need the apostrophe to plot the 2D data correctly
```

Plot the output for each voxel. How different are the averages of 10 trials between the high signal and low signal voxels?

3.2. Plot the HDR as a function of trial number

Here you will create graphs that show how the hemodynamic response changes with trial number. The function "average_epochs" has been written for this exercise. It is in the same directory as the other scripts used today. You should select some numbers of trials from 1 to 96 that you think will illustrate the changes in the HDR. Think about how noise changes with number of trials when choosing your values. An example of the script is below, which would find an average HDR for 1, 2, 3, and 4 trials. **Test the effects of signal averaging on all of your voxels.**

```
output = average_epochs(data_visual,[1,2,3,4]);
```

How does the HDR change as you increase your experimental power? What happens to the baseline period? What happens to the task period? How are the results different between high and low signal voxels? Plot the results using Excel and print out the figures compactly. Use the toexcel command to send the output to excel. Print all of the curves for one voxel on one figure (4 total printouts).

3.3. Creating a script for calculating correlations

In this step, you will write a short script to calculate, for samples of 1 to 96 trials, how the correlation with a reference waveform changes with signal averaging. Examples of the types of commands that you will need are provided, but you will have to write the necessary script yourself. You can load the variable weights_matlab (same directory) to use as a correlation template, or you can create one yourself.

First create a new blank script: edit trial_averaging
Your script will need to do the following:

1) Create a variable to store your output: output_data = zeros(96,1)
2) Loop from 1 to 96 trial samples (using the for command)
3) Create a sample of N trials from the 96 total (using the sort_epochs function)
4) For each sample, determine the average HDR (using the mean command)
5) For each average HDR, determine the correlation matrix (using corrcoef)
    a. correlation_matrix = corrcoef(mean_data,weights_matlab);
6) Select the correlation coefficient from the correlation matrix
    a. output_data(i)=correlation_matrix(1,2);

3.4. Calculating effects of trial averaging upon significance

You should run your script on each of the voxels from above. **How does the correlation value change as you increase the number of trials averaged? What is different about your high and low significance voxels?**

3.5. Challenge exercise

Now, go back to Exercise 1 and find the voxel that is significant at t>2.0 and has the smallest signal change (percentage change over baseline) that you can measure. **Repeat the exercise from 3.4. How do the results differ?** [Hint: you may want to use the combination of two find commands to locate this voxel]

Exercise 4: Bonus challenge

In Exercise 1, you selected half of your runs for an analysis. You can now go back and analyze the other ½ of your runs. **How similar are the results from the two split-data analysis? Can you quantify how similar are the spatial patterns of data?**