

Week 3 Exercises

Last week you began working with MR data, both in the form of anatomical images and functional time series. This week we will discuss some concepts related to the idea of fMRI data as sets of numbers that can be manipulated computationally. We will also introduce the concept of "statistical maps" (or "overlays"). We will be working with statistical maps throughout the course, so we are introducing them early in the laboratories.

We will talk about automated ways of loading data, processing, and viewing data using the **BIAC XML header** format ("*.bxh"). You will use the same basic viewing tool, Showsrs2, but with a new way of loading the data.

Preparing for the exercise

0.1. Setting up the system path to allow use of BIAC XML headers (BXH).

Depending on the computer(s), you may only have to do this once.

- a) Press <windows> + <break> simultaneously to bring up the "system properties" box.
- b) IN XP: Click on the "Advanced" tab. Click on "Environment Variables". Click New
IN NT: Click on Environment
- c) For variable name, type **path**. For variable value, type **p:\matlab\biac\lib**
- d) Exit from system properties. Restart MATLAB if you have it open.

0.2. Mapping the server's hard drive

Remember that you need to have the drive [\\Huxley\data](#) mapped for these examples to work well. It is suggested that you map this directory to the drive letter S: , and that you check the "reconnect on log-in" box so that you do not have to map it each time.

0.3. Adding BXH scripts to MATLAB path

This has to be done each session that you want to use the newest version of the BIAC scripts, which use the BXH format. Note that these scripts are *alpha* versions, and are not yet in final release.

```
addpath p:\user_Scripts\alpha\
```

Exercise 1: Loading anatomical data using BXH

1.1 Create a directory and move the dataset

You will be working with a high resolution anatomical dataset, for which you will create a BXH configuration file. This data consists of an axial T₁-weighted image with dimensions 256 * 256 * 86, and voxel size 0.9375mm * 0.9375mm * 1.5mm. Slice #1 is the bottom of the brain; slice #86 is the top of the brain. It is saved in the “volume” format, although it has a Analyze format header already. You will need to **create a directory** for your group under Class.01\Students. You will not need to copy any files to your new directory in this lesson; you will just use it for storing the headers that you create. Then **change to that directory** in MATLAB.

1.2 Create a BXH file for the dataset

The file that you will be working with is:

\\Huxley\data\class.01\examples\anatomical_images\3D_SPGR_Full_Brain\3D_SPGR_a.img

To create the BXH file, you will need to use the command **bxhabsorb**. If it does not run, then you need to add the path information from above. Once you type **bxabsorb**, a graphic user interface will pop up. You should select the file above; remember that you may need to change file type in the dialog to make it visible. You will need to put in the voxel size and file type (of volume; at top right). Leave “oblique” unchecked for now. Save the BXH file in your directory, and close the window.

1.3 Displaying the data directly using BXH

One of the key features of the BXH format is that the header information, once specified, can be used to automatically load and view the data. That is, you do not need to describe the data each time that you load or view them. To read the data, just type:

anat = readmr

Select the BXH file that you just created and click OK. Now you have loaded in to MATLAB a structure that contains two fields: anat.data, which has the actual image data; and anat.info, which has the configuration info. You can look at the configuration info by typing **anat.info** at the MATLAB command prompt. Note that you do not need to type in any additional information to load the file, once the header has been created.

To display this image using showsrs2, type

showsrs2(anat)

Showsrs2 automatically determines the correct configuration for displaying the data from the file itself, so you do not have to flip or resize the data.

Exercise 2: Understanding overlays

In this next exercise, you will work with data collected in a biological motion paradigm. Subjects watched three types of motion (of cartoon figures): mouth, eye, and hand. The data are stored in the directory [S:\Class.01\Examples\Biological Motion](#). The AVI files in this directory show the stimuli that were used. Each of the #####_overlay files indicates the value of a statistical test (t) at each voxel.

You will create the BXH files in your group directory, but leave the original data in the examples directory.

2.1 Create BXH files for each image type

You will need to create four BXH files. Each should point to the data in the examples directory.

- 1) base.bxh : for the “base.img” file.
- 2) mouth.bxh : for the mouth_overlay.img file
- 3) eye.bxh : for the eye_overlay.img file
- 4) hand.bxh : for the hand_overlay.img file

All have parameters:

- a. Voxel dimensions: 90 * 108 * 78
- b. Voxel size: 1.5mm * 1.5mm * 1.5mm
- c. Data type: float
- d. Orientation: Axial (S=>I)

After you have created these files, load in the data to MATLAB. You should end up with four variables (base, mouth, eye, and hand). Note that you can load the data, once a BXH file has been created, either by using the readmr interface or by just typing the following (if ‘base.bxh’ was in your current directory).

```
base = readmr('base.bxh'); % repeat for the other files
```

2.2 Display the data using Showsrs2

To display the data, you can just type:

```
showsrs2(base, mouth, eye, hand)
```

Q1. Describe the data. What do you see as you move around the brain?

Q2. What do you think that the different values in the bottom right window represent? Are they generally similar or different across the three conditions?

2.2 Understanding overlays

These color maps are often called “overlays” because they are laid on top of the original anatomical data. **The single most important thing to understand about such**

overlays is that they represent statistical tests of whether the data are consistent/inconsistent with an experimental hypothesis. They are not “snapshots” of brain activity. To illustrate this point, look at the data in showsrs2, with all thresholds set to the default values (as loaded automatically).

Q3. How many discrete areas are active in the brain? Give a rough estimate of the number of large clusters that you see. Also, about what percentage of the brain is active?

Now, lower all of your thresholds for the different overlays to a lower limit of 1.0. To do so, you can click on the W+L button and select the different overlays using the pull-down menu at upper right.

Q4. After adjusting the threshold downward, how did the brain activity change from the previous threshold?

Finally, raise all of the thresholds to a lower limit of 6.5.

Q5. After adjusting the threshold upward, how many areas are now active? What is different about the activity?

Note that the same brain overlays were used in every case. The only difference was in the statistical value that we called “significant”.

2.3 Quantifying the effects of threshold

We can very quickly count how many voxels have significance values above different threshold levels in MATLAB. Suppose that you want to count how many voxels have a t value > 0.5 , a t value > 1.0 , a t value > 1.5 , and so forth. You could define a set of such steps (0.5, 1.0, 1.5, ..., 7.5, 8.0) using the command:

```
steps = [0.5: 0.5 : 8];
```

Then you could set up a place to store data:

```
a=zeros(length(steps),1);
```

Finally, you can quickly count how many voxels had significance values greater than each step. (You can plot the output using the **figure** and **plot** commands).

```
for i = 1:length(steps);a(i)=length(find(mouth.data>steps(i)));end
```

Q6. How does the number of active voxels change with significance value?

BONUS Exercise: You could also plot how many voxels are within small ranges (e.g., from 0 to 0.2, from 0.2 to 0.4, etc.) in order to map out the distribution of significance values across voxels.

2.4 Finding the maximum and minimum values

You should next find points in the brain with the maximum values for each overlay (note that there may be more than one point with a given value). Hint: adjust the thresholds in the configuration window so that you display one overlay at a time, and adjust the range for that overlay so that you only see the top of the range. The movement keys (A&D, W&S, and Q&E) may be useful. Write those values below:

Q7. What are the coordinates of the maximum significance values, estimated visually:

eye_overlay: maximum (X,Y,Z): _____ t = _____

hand_overlay: maximum (X,Y,Z): _____ t = _____

mouth_overlay: maximum (X,Y,Z): _____ t = _____

2.6 Using transparencies

Using the transparency function on the "W+L" screen, determine whether each of your maxima are found in gray matter or white matter. Put a G or W next to the points listed above.

2.7 Check your answer

You can also quickly find out the minimum and maximum values in each image:

mm = minmax(mouth.data)

To determine the X, Y, and Z coordinates of these values, use the following:

mm_min = find(mouth.data == mm(1))

mm_max = find(mouth.data == mm(2))

[x,y,z]= ind2sub(size(mouth.data), mm_min)

[x,y,z]= ind2sub(size(mouth.data), mm_max)

Q8. What are the coordinates of the minimum and maximum significance values, determined in MATLAB? What are the actual t values?

eye_overlay: min: _____ t = _____ max: _____ t = _____

hand_overlay: min: _____ t = _____ max: _____ t = _____

mouth_overlay: min: _____ t = _____ max: _____ t = _____

Exercise 3: Working with Time Series of data

3.1. Creating BXH files for time series

A "time series" of data is a sequence of similar images showing changes in the brain over time. You are going to work with "averaged epochs", which indicate the changes in the brain time-locked to the occurrence of the different stimuli. These data consist of 12 time points, from 3 time points before (6s) through 8 time points after the

stimulus (16 sec). The data are saved in the same directory as files in the form :
[Eye_0001.img](#).

You will need to create BXH files for each of the three time series. The only thing that you need to do differently is to select “use wildcard read”; if you then select one file in a series, it will associate that BXH file with all of them. The examples below will assume that you use the file convention: [eye_all.bxh](#) . When you are finished creating the BXH files, 1) load all three time series into MATLAB and 2) display them in showsrs2.

[showsrs2\(base, mouth_all, eye_all, hand_all\)](#)

3.2. Changing the colormap limits

You will need to change the values displayed on the colormap, as we will now be dealing with absolute change instead of t values. Change the lower limit of each overlay to 2.0 and the upper limit to 3.5.

3.3. Looking at time series of data

Find the points that you identified as being active in all three conditions. What does the time course of activation look like at those points? You can view the data in the window at bottom right, and/or play the data using the controls at bottom left.

Q9. Write a short description of the time course of activity at the “most active voxels” in the space below.

3.4. Finding the peak response

Q10. At what point does the brain response peak? Is this similar for all of the conditions? Is it similar across all of the points? Write a short description of your findings in the space below.

3.5. Looking at the spatial pattern of activation

Q11. How does the spatial pattern of activation change across time points? At what time point is there the most activation? Write down your findings in the space below.

Summary:

Q12. Write a general summary of the data below. You looked at (real) data showing patterns and time courses of activity when the subjects were watching eye, hand, and mouth movements. What brain areas were activated by these stimuli? How did the activation pattern in those areas change over time? How similar were the activation patterns for the different conditions?

Questions:

Q1. Describe the data. What do you see as you move around the brain?

Q2. What do you think that the different values in the bottom right window represent? Are they generally similar or different across the three conditions?

Q3. How many discrete areas are active in the brain? Give a rough estimate of the number of large clusters that you see. Also, about what percentage of the brain is active?

Q4. After adjusting the threshold downward, how did the brain activity change from the previous threshold?

Q5. After adjusting the threshold upward, how many areas are now active? What is different about the activity?

Q6. How does the number of active voxels change with significance value?

Q7. What are the coordinates of the maximum significance values, estimated visually:

eye_overlay: maximum (X,Y,Z): _____ $t =$ _____

hand_overlay: maximum (X,Y,Z): _____ $t =$ _____

mouth_overlay: maximum (X,Y,Z): _____ $t =$ _____

Q8. What are the coordinates of the minimum and maximum significance values, determined in MATLAB? What are the actual t values?

eye_overlay: min: _____ $t =$ _____ max: _____ $t =$ _____

hand_overlay: min: _____ $t =$ _____ max: _____ $t =$ _____

mouth_overlay: min: _____ $t =$ _____ max: _____ $t =$ _____

Q9. Write a short description of the time course of activity at the “most active voxels” in the space below.

Q10. At what point does the brain response peak? Is this similar for all of the conditions? Is it similar across all of the points? Write a short description of your findings in the space below.

Q11. How does the spatial pattern of activation change across time points? At what time point is there the most activation? Write down your findings in the space below.

Q12. Write a general summary of the data below. You looked at (real) data showing patterns and time courses of activity when the subjects were watching eye, hand, and mouth movements. What brain areas were activated by these stimuli? How did the activation pattern in those areas change over time? How similar were the activation patterns for the different conditions?