

## Week 5 Exercises

This week's exercises will introduce concepts related to analysis of functional data. You will work with a complete functional dataset collected on the BIAC 4.0T scanner. The dataset contains a single session from one subject who ran in two types of designs: 1) a blocked-design with a flickering checkerboard (12s on, 12s off, etc.) and 2) a blocked-design with moving bars (12s up, 12s off, 12s down, 12s off, etc.). Runs 1-5 were the flickering checkerboard, runs 6-8 were the moving bars.

The custom functions needed are `mrtest`, `tstatprofile`, and `showsrs2`.

### Exercise 1: Understanding the basic concepts of functional data

#### 1.1 Data Organization

The data are organized into the following categories, from largest to smallest: examination, series, runs, volumes, slices, and voxels.

**Examination:** Each time that a session is started on the MR console, a unique examination number is generated. These numbers started from 0 on the 1.5T and from 40000 on the 4.0T. Note that one subject may have more than one examination number in rare cases, such as if the scanner computer crashed or we had to interrupt the session for some reason.

**Series:** Each time a different pulse sequence is run within an exam, it is given a new series number. So we may run one series of a sagittal SPGR scout, a second of an axial spin-echo anatomical, and a third of a spiral gradient echo functional.

**Runs:** If a single sequence is conducted more than once in succession, each time generates a new run. Mostly, this applies to functional series, where we will do perhaps 8 runs of the same task.

**Volume:** Each run may contain one or more volumes. For a functional run, there may be 300 volumes or more. Note that each volume corresponds to the data acquired in a single TR.

**Slice:** Within each volume, there will usually be more than one slice. Typical full-brain functional imaging protocols acquire 22-30 slices. Sometimes fewer slices will be acquired if a particular part of the brain is of interest and having a short TR is desirable.

**Voxel:** Within each slice, there are a large number of voxels (e.g., 4096 for a 64\*64 matrix).

#### 1.2. Data Files

All of the data that you will use in this exercise is in the following directory. **Change to this directory in Windows NT Explorer and look at the files.** (Note: this assumes that you have [\\Huxley\Data](#) mapped to the drive letter S:\)

S:\Class.01\Examples\Functional\_Data\Flash\_Motion\

##### **Types of Data Files:**

**pfiles:** The functional data comes from the scanner stored in "P files". Each P file contains the raw kspace data from an entire run. The files are usually very large (>100mb). The P files are not included in these directories; they are archived to tape immediately after being collected.

**func:** The P files are converted into functional data organized by runs. These runs are automatically labeled in the following format: run<series>\_<number>, i.e., run004\_01. Each image is saved as V0001.img, V0002.img, etc.

**anat:** The anatomical images are stored in separate series. Note that normally these images are stored in DICOM format, but here they have been rewritten into Volume format (to remove identifying header information). We will not use the anatomical images in this exercise.

There is also a directory called **stimuli** that shows the stimuli and display files used for this experiment.

### **1.3 Details of the sample dataset.**

As mentioned above, this dataset consists of two visual experiments done with a single subject: flashing (checkerboard) and moving (bars). All stimulus files are in the "stimuli" subdirectory. All functional images are 64\*64\*20, with 3.75\*3.75\*5mm voxels. Both datasets used a 1s TR, with 4 discarded and deleted acquisitions (disdaqs).

#### **1.3.1. Flashing experiment**

In the flashing experiment (runs 1-5), the subject saw 12s of a counterphase flickering radial checkerboard alternating with 12s of a static checkerboard. The first flicker started at time point 9 in the volume series, and then repeated for 12s blocks (12s on / 12s off / 12s on / etc.) until the end of each run. So, the first flashing block went from time points 9-20, the second from 33-44, etc.

#### **1.3.2. Motion experiment**

In the motion experiment (runs 6-8), the subject saw 12s of upward motion, then 12s of static bars, then 12s of downward motion, then 12s of static bars, etc., repeating until the end of the run. The first flicker started at time point 13 following the disdaqs.

## **Exercise 2: Using mrtest**

In this exercise, you will use the analysis program mrtest to do a blocked-design analysis of these data. You need to create two files, a header and a paradigm, and then run mrtest. Create a directory under "students" to put these files for your group. Note that you will need separate subdirectories for the flashing and motion data if you analyze both.

### **2.1. Creating a header file**

You should copy the "sample\_mrtest\_header.txt" file to your directory and rename it. You will need to modify the file so that it has the right values for each variable. Make sure to have different output directories for the flash and motion data.

### **2.2. Creating a paradigm file**

You should create a paradigm file that consists of a series of time points that correspond to the points you want to include in each condition (e.g., flash vs. static). For each TR in the time series of volumes, you should either have a 1 (for task), a 2 (for non task), or a 0 for points that don't go into either category. It is probably easiest to create these using Excel, and then paste the results into a text file.

There are two things that you should consider when creating your paradigm file:

- 1) Introducing a delay from the onset of the block to the label of "task" activation
- 2) Inserting extra spacing between the blocks to account for transitions between them

### **2.3. Running mrtest**

To run mrtest, simply start Matlab and type **mrtest**, and then select your header file. If MATLAB does not find mrtest, just add the appropriate directory to your path:

`addpath P:\MATLAB\BIAC\MR`

The output of `mrtest` is a file called `Map_1-2.T`; this file is a statistical map showing the t-statistic for each voxel based on your paradigm file. You can view this in `showsrs` or `showsrs2`; use a single functional image from one of the runs as a base image.

## Exercise 3: Using `tstatprofile`

In this exercise, you will use the analysis program `tstatprofile` to do an event-related analysis of these data. You need to create three files, a header, a paradigm, and a template, and then run `tstatprofile`. Use the directory under "students" to put these files for your group. The output of `tstatprofile` will not overwrite that of `MRtest`

### 3.1. Creating a header file

You should copy the "sample\_tstat\_header.txt" file to your directory and rename it. You will need to modify the file so that it has the right values for each variable. Make sure to have different output directories for the flash and motion data.

### 3.2. Creating a paradigm file

You should create a paradigm file that consists of a series of time points that correspond to the points you want to include in each condition (e.g., flash vs. static). This paradigm file has a slightly different organization. Here, you should indicate where events happened in the time series by putting a 1 in the paradigm file at the TR where each event happened. If you are analyzing the motion data, just group both types of motion together and put a 1 at the onset of each motion block.

### 3.3. Creating a template file

The template file is a series of weights that indicate your hypothesis about the possible hemodynamic response following each stimulus. In this case, with the stimulus block lasting 12 TRs, you may want to include something like 5 TRs before the stimulus through 12 TRs after (total of 18 time points). Remember what we have talked about regarding the general form of the hemodynamic response when creating your template file.

### 3.4. Running `tstatprofile`

To run `tstatprofile`, simply start Matlab and type `tstatprofile`. You will then need to select your header file. If MATLAB does not find `mrtest`, just add the appropriate directory to your path:

`addpath P:\MATLAB\BIAC\MR`

`Tstatprofile` provides a large number of different types of output. See the `tstatprofile` instructions in the education section of the website for details. You should look at the different types of output to see what they represent.

## Exercise 4: Comparisons of output results

Note that these tools give you very different output. You should be able to answer the following questions (use a separate sheet):

**4.1. How does the `mrtest` output change depending upon changes in the paradigm file? If you shift the position of blocks in your paradigm file up or down (i.e., earlier or later in time), what happens?**

**4.2. What is the optimal hemodynamic delay?**

**4.3. What is the optimal spacing between blocks?**

**4.4. What different types of output does `tstatprofile` give?**

Names of students: \_\_\_\_\_  
fMRI Graduate Class

Week 5

- 4.5. Why would you use each of the different types of outputs?**
- 4.6. What is the form of the best template file that you have created?**
- 4.7. How do the optimal mrtest and tstatprofile results compare?**
- 4.8. Why would you use one tool instead of the other?**