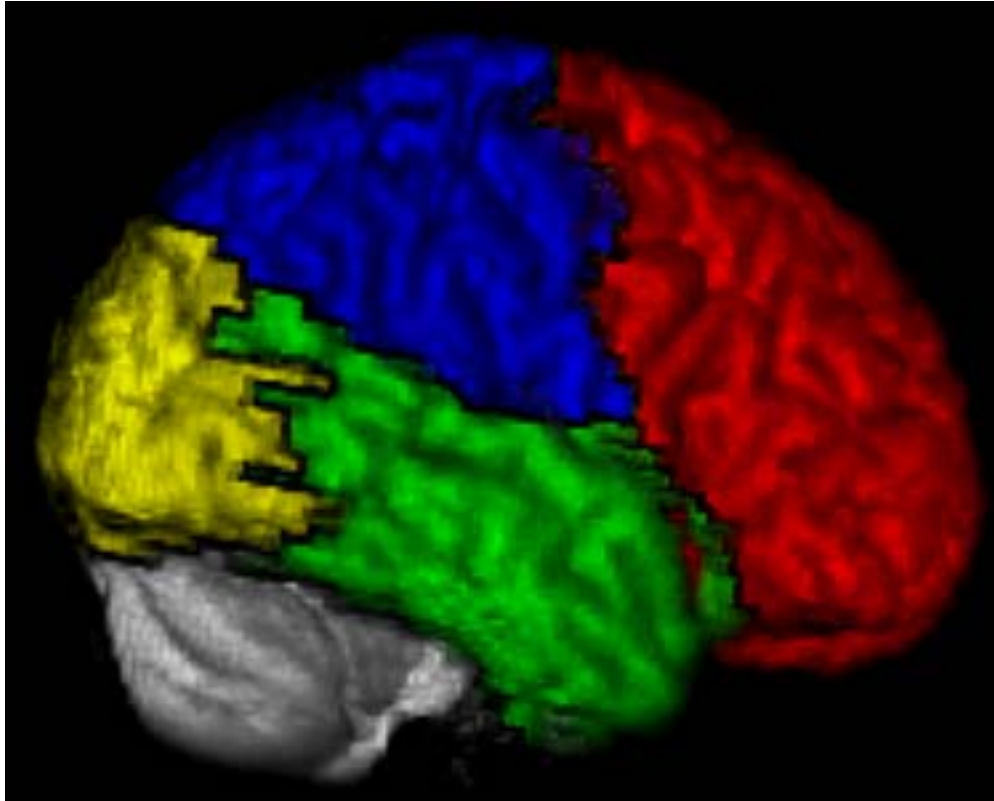


WFU SPM5 PIPELINE User Manual v1.0

ANSIR Laboratory
Wake Forest University School of Medicine



WAKE FOREST
UNIVERSITY

SCHOOL *of* MEDICINE

1. Software Description, Installation and Setup.....	3
1.1 Description.....	3
1.2 Downloading the Software	3
1.3 Program Installation.....	3
1.3.1 System Requirements.....	3
1.3.2 Setting up the directories	3
1.3.3 Compiling required component: DCMTK.....	4
1.3.4 Configuring the SPM5 Pipeline.....	5
1.3.5 Considerations before compiling and using MoJoe.....	7
1.3.6 Compiling optional component: FLTK (required for MoJoe).....	7
1.3.7 Compiling optional component: MoJoe.....	8
1.3.8 Configuring MoJoe, if MoJoe is to be installed.....	8
1.3.9 Optional Transfer of MoJoe to a MRI scanner	11
1.3.10 Final installation and setup of MoJoe, if MoJoe is to be installed	11
1.3.11 Reference Functions.....	12
1.4 Compatibility	15
1.5 Referencing the software	16
2. Using the Software.....	16
2.1 Using MoJoe	16
2.1.1 MoJoe: Capture.....	16
2.1.2 MoJoe: Extract	17
2.2 Autoprocessing	18
2.2.1 Autoprocessing from MoJoe.....	18
2.2.2 Autoprocessing P-files from the Command line.....	18
2.2.3 Autoprocessing DICOM images from the Command line (DICOM images that do not have the paradigm number set).....	18
2.2.4 Autoprocessing DICOM images from the Command line (DICOM images that have the paradigm number set).....	19
2.2.4 Autoprocessing 4D files from the Command line	19
2.3 Reprocessing and Error Recovery	20
2.3.1 cleanrun.....	20
2.3.2 rerun	20
2.3.3 rerun_local	20
2.3.4 quick_start.....	20
3. Technical Notes	21
3.1 Transferring of fMRI analysis to PACS.....	21
3.2 WFU Grid Distribution Tool	21
References.....	22

WFU SPM5 Pipeline version 1.0 User Manual Joseph Maldjian, MD

1. Software Description, Installation and Setup

1.1 Description

The WFU SPM5 Pipeline is a fully automated method for the processing of fMRI data using SPM. It is fully automated from the point of data acquisition at the MRI scanner. It incorporates tools for automated data transfer, archiving, real-time SPM5 batch script generation with distributed grid processing, automated error-recovery procedures, full data-provenance, email notifications, optional conversion back to DICOM (Digital Imaging and Communications in Medicine), and picture archiving and communications systems (PACS) insertion. The architecture allows for an infinite number of easily definable analyses that are fully automated from the point of acquisition. This pipeline was developed in the ANSIR Laboratory at the Wake Forest University School of Medicine. Questions can be referred to maldjian@wfubmc.edu.

1.2 Downloading the Software

The WFU SPM5 Pipeline can be obtained at www.ansir.wfubmc.edu.

1.3 Program Installation

1.3.1 System Requirements

- ksh (Korn Shell Script)
- csh or tcsh (C Shell Script)
- mailx
- Sun Grid Engine 5 or 6
- Offis's Dicom Toolkit (dcmtk) 3.5.4 (included, needs to be compiled)
- Fast, Light Toolkit (fltk) 1.1.2 (included, needs to be compiled)
- Cdtools 1.10 (included, may need to be compiled)
- FTP and SSH server installed on at least one grid node

1.3.2 Setting up the directories

The WFU SPM5 Pipeline is designed for a grid computing environment. It is strongly recommended that pipeline files be placed on a filesystem that is accessible to all computers that are on the grid. While it could be individually installed on each grid computer, this is not recommended. Data has to be stored on a filesystem that is accessible to all nodes in the grid.

Change to the parent directory of where the pipeline is to be installed. Place the downloaded tar.gz file in this directory. Uncompress and extract the files in the archive:

```
tar -xzf WFU_Pipeline.tar.gz
```

The following directories (and many further subdirectories) will be created:

WFU_Pipeline/	The root directory of the pipeline
WFU_Pipeline/WFU	Mainly Matlab code used in autoproccessing
WFU_Pipeline/bin	Scripts useful for processing
WFU_Pipeline/doc	Documentation
WFU_Pipeline/external	Source code for needed libraries
WFU_Pipeline/fmri	Scripts used in autoproccessing
WFU_Pipeline/mojoe	source code for MoJoe
WFU_Pipeline/ref_functions	the location of assorted reference functions and setting for MoJoe

If the sample_data has been downloaded, place it in the same location as the WFU_SPM5_Pipeline.tar.gz and uncompress it as well.

```
tar -xzf WFU_Pipeline_Sample_Data.tar.gz
```

The following directories will be created:

WFU_Pipeline/poem052_08_28_08	A sample study
WFU_Pipeline/testbed	a sample “MRI” directory structure for testing MoJoe

Next, set the environmental variable WFU_PIPELINE. This will depend on you default shell. For a bash type shell:

```
cd WFU_Pipeline  
export WFU_PIPELINE=`pwd`
```

For a csh type shell:

```
cd WFU_Pipeline  
setenv WFU_PIPELINE=`pwd`
```

Note: The tick marks above are back ticks (found usually above the tilde [~] key).

To check that the variable has been set correctly:

```
printenv | grep WFU_PIPELINE
```

1.3.3 Compiling required component: DCMTK

The WFU SPM5 Pipeline requires access to several executables found in Offis’s Dicom Toolkit (<http://dicom.offis.de/dcmtk>). DCMTK may be compiled/installed as root and placed in one of the standard bin directories. Alternatively, it may be compiled/installed as an unprivileged user and placed in alternative directory. If this method followed, the WFU SPM5 Pipeline user(s) will have to have this directory added to their path.

Note: the MoJoe compiling script will expect DCMTK to be located spm5_pipeline/dcmtk-3.5.4

Uncompress and extract the files in the dcmtdk archive.

```
cd ${WFU_PIPELINE}  
tar -xzf external/dcmtdk-3.5.4.tar.gz
```

Configure the package to your system. This documentation will refer to an install in the /opt directory. To use the standard linux bin directories, remove the --prefix=/opt option.

```
cd dcmtdk-3.5.4  
./configure --prefix=/opt
```

Next, make the DCMTK package and install it. Do not run 'make clean' if planning to use MoJoe.

```
make  
make install
```

Note: DCMTK may now need to be added to your path. If `which dcmdump` does not find the command, then this must be done.

1.3.4 Configuring the SPM5 Pipeline

Depending on the default shell for your users, several files will need to be created/modified for the WFU SPM5 Pipeline to run. On our test system with the default shell being bash, the following are required files and the appropriate settings:

.bashrc file (The default shell)

```
# /etc/skel/.bashrc  
#  
# This file is sourced by all *interactive* bash shells on startup,  
# including some apparently interactive shells such as scp and rcp  
# that can't tolerate any output. So make sure this doesn't display  
# anything or bad things will happen !  
  
. /opt/sge6/default/common/settings.sh  
  
#for dcmtdk  
export DCMDICTPATH=/opt/lib/dicom.dic:/opt/lib/private.dic  
  
#For spm5_pipeline  
#This should go before the interactive shell test exits  
export HOST=$(hostname)  
export WFU_PIPELINE=/home/bwagner/spm5_pipeline  
export WFU_PATH=${WFU_PIPELINE}/WFU  
export WFU_SPM5=/home/bwagner/spm5  
export MATLABPATH=${WFU_PATH}:${WFU_SPM5}  
  
# Test for an interactive shell. There is no need to set anything  
# past this point for scp and rcp, and it's important to refrain from  
# outputting anything in those cases.  
if [[ $- != *i* ]] ; then  
    # Shell is non-interactive. Be done now!  
    return  
fi
```

.cshrc file (Default shell for SGE)

```
source /opt/sge6/default/common/settings.csh

#for dcmtk
setenv DCMDICTPATH /opt/lib/dicom.dic:/opt/lib/private.dic

#For spm5_pipeline
#This should go before the interactive shell test exits
setenv WFH_PIPELINE /home/bwagner/WFU_Pipeline
setenv WFU_PATH ${WFU_PIPELINE}/WFU
setenv WFU_SPM5 /home/bwagner/spm5
setenv MATLABPATH ${WFU_PATH}:${WFU_SPM5}
```

The various elements that have been defined are:

DCMDICTPATH	The location of the dictionary files needed by the DCMTK.
WFU_PIPELINE	The location of the WFU_SPM5_PIPELINE.
WFU_PATH	The location of the WFU directory in the WFU SPM5 Pipeline.
WFU_SPM5	The location of the SPM5.
MATLABPATH	Additional paths that Matlab should load when it starts. This should include the WFU_PATH and WFU_SPM5 path. The \${WFU_PATH} and \${WFU_SPM5} references the variables already set.

Finally, copy the global.site.ksh.template file to global.site.ksh.

```
cd ${WFU_PIPELINE}/WFU
cp global.site.ksh.template global.site.ksh
```

At a minimum, edit global.site.ksh file and configure the following settings:

refdir	Full path of the reference functions.
su_email	Super user email. Place the username of this person after the grep command.
email_address	Change the domain name as appropriate.
rsh_flag	If set to 1, then the mailserver and timekeeper may be used for specified task. Otherwise the local machine will.
mailserver	Hostname, FQDN, or IP of the main computer to send emails from.
mail_flag	If set to 1, then the mailserver will be used to send emails. If set to 0, then the computer which runs the script will be used. This is only in effect if the rsh_flag is set.
timekeeper	Hostname, FQDN, or IP of the computer used to get the date and time.
batch_display	Hostname, FQDN, or IP of the computer to push all displays from Matlab to. This computer must have an X server installed and 'xhost +' issued on it.
batch_matlab	Full path to the Matlab execution script along with arguments used at startup. It is recommended that the options '-nosplash' and '-'

nodesktop' be used. The '-nojvm' option can also be used, but may not be supported in newer Matlabs.

Running the included sample study:

First, edit the `/${WFU_PIPELINE}/ref_functions/paradigm_lookup_table`. Replace the text `'/home/bwagner/spm5_pipeline'` with the appropriate path to the pipeline. Then see section: **2.2.4 Autoprocessing DICOM images from the Command line (DICOM images that have the paradigm number set).**

1.3.5 Considerations before compiling and using MoJoe

MoJoe's intended use is on GE LX MRI scanners. The MoJoe that is about to be built should run on the computer it has been compiled on. However, as scanners have very specific libraries and utilities available, it may not run on the scanner. If this occurs, try to find/make a computer with a similar configuration as the scanner. Then follow these instructions again for making the DCMTK, FLTK, and MoJoe. As a starting point, please check the gcc version on the compiling computer and scanner by typing 'gcc -version.'

IMPORTANT NOTE:

Autoprocessing will stop after dicom conversion and 4D image creation if dicom field 0019,10e0 is not set to a valid reference function. The method of setting this field varies by scanner and scanner usage policies.

If MoJoe is not to be installed, please skip to section: **1.3.11 Reference Functions.**

1.3.6 Compiling optional component: FLTK (required for MoJoe)

The MoJoe requires access to several libraries found in the DCMTK and the Fast, Light Toolkit (<http://www.fltk.org>). As the version of FLTK used with MoJoe is not the most recent, it is recommended to compile FLTK in its source directory and not install it to the system directories.

Note: The MoJoe compiling script will expect FLTK to be in located `spm5_pipeline/fltk-1.1.2`

Uncompress and extract the files in the fltk archive.

```
cd ${WFU_PIPELINE}  
tar -xzf external/fltk-1.1.2-source.tar.gz
```

Configure the package to your system.

```
cd fltk-1.1.2  
./configure --prefix=${WFU_PIPELINE}/fltk-1.1.2
```

Make the FLTK package. Do not run ‘make install’ or ‘make clean’ from the fltk-1.1.2 directory.

```
make
cd FL
make install
```

1.3.7 Compiling optional component: MoJoe

MoJoe has several executables in the moJoe directory that need to be built. To ease this process, a Korn Shell Script has been written to compile all the needed parts.

```
cd ${WFU_PIPELINE}/moJoe
./makeall_moJoe
```

1.3.8 Configuring MoJoe, if MoJoe is to be installed

MoJoe has several configuration files found in two locations: `${WFU_PIPELINE}/ref_functions/moJoe_users/generic` and `${WFU_PIPELINE}/moJoe`. The first directory contains information on the computing grid. The second contains scanner settings and how to communicate with the grid.

The following files tell moJoe how your grid is set up and are found in the `${WFU_PIPELINE}/ref_functions/moJoe_users/generic` directory:

disks.txt	Locations of assorted drives that MoJoe may send to.
disks2.txt	Same as above, but with a Windows mapping column.
userdisk.txt	Defines which drives user’s studies should default to.
userlist	A list of users and the studies that they are associated with.

When the MoJoe setup script is run, these files are transferred to the scanner. Additionally, the userlist file is updated each time MoJoe is run. MoJoe also has setting specific to the computer (scanner) it is running on in the **settings.ksh** file of the moJoe directory. The settings found in these assorted files are detailed below.

```
${WFU_PIPELINE}/ref_functions/moJoe_users/generic/disks.txt
#disk      mount_point  host_disk
local      /studies     elfido:/studies
local2     /studies2    elfido:/studies2
```

```
${WFU_PIPELINE}/ref_functions/moJoe_users/generic/disks2.txt
#disk      mount_point  host_disk      pc_mount
local      /studies     elfido:/studies //elfido/studies
local2     /studies2    elfido:/studies2 //elfido/studies2
```

The **disks.txt** and **disks2.txt** should be nearly identical, with the latter containing an extra column. These are multiple-space (or tab) delimited files. The first column is the label that MoJoe will display to the user as the drive the data will be sent to. The second column is the mount point to which node computer will find this drive. The third column is the NFS mounting

information MoJoe needs in order to data. The fourth column (disk2.txt only) is the Windows share name for the disk location. This is so that the emails sent out automatically can have links to the data for Windows users.

```
${WFU_PIPELINE}/ref_functions/mojoe_users/generic/userlist
```

```
#username,email,enableitem,enablesub,subdir1,subdir2,subdir3,...  
clinical,maldjian@wfubmc.edu,1,1,pas1,dti,fmri,misc,qa,qa/acr  
-----,0  
bwagner,bwagner@wfubmc.edu,1,1,test1,test2,test3
```

The **userlist** file contains a list of users, some settings, and the studies that the users have permission to write to. There should be no space in between the comma delimited settings. It is recommended that the username be the PI for the study as directory that the data will be written to will be in the form of drive/username/study. The second line is a separator to differentiate the clinical user from the research users below it.

```
${WFU_PIPELINE}/ref_functions/mojoe_users/generic/userdisk.txt
```

```
#username defaultdisk  
bwagner local  
bwagner:test1 local2  
bwagner:test2 local2
```

The **userdisk.txt** file defines which drives data for a particular user will be stored on. The format is username:study drive. The username is the same as found in the userlist file, as is the optional study. The default disk should reference a label in the disk.txt file. In the above sample, all user bwagner's data will go to the "local" disk, except for those in the test1 and test2 studies. Data for test1 and test2 will go to the "local2" disk.

***Note:** The disk/username directory must exist and have permissions set so that the sshuser (see below) can write to these directories before using moJoe. For example, the above referenced user should have the following directories already created: /studies2/bwagner/test1, /studies2/bwagner/test2, and /studies/bwagner/test3.*

The only file in **\${WFU_PIPELINE}/moJoe** to configure is the settings.ksh.template file. First copy the template to settings.ksh.

```
cd ${WFU_PIPELINE}/moJoe  
cp settings.ksh.template settings.ksh
```

Mojoe/settings.ksh

```
sshserver=elfido.medeng.wfubmc.edu
sshservername=${sshserver} # can be used to shorten the server, or a alpha name if the
above is an IP
sshuser=mojoe_ssh
sshpass=junk2007

#this is the location of the reference functions directory on the FTP/SSH server
ref_dir=/home/bwagner/spm5_pipeline/ref_functions

#these are the locations of assorted directories in the pipeline
fmri_dir=/home/bwagner/spm5_pipeline/fmri
bin_dir=/home/bwagner/spm5_pipeline/bin

#billing info
billEmail=bwagner@wfubmc.edu
billRemoteDir=/studies/mojoe_bill

#monitoring info
monitorEmail=bwagner@wfubmc.edu
monitorRemoteDir=/studies/mojoe_monitor

#this is the location of this mojoe installation on a test box
#on a scanner, this should be in a directory called mojoe off the user directory and
all lines below should be commented out.
PIPELINE_DIR=/home/bwagner/spm5_pipeline
MOJOE_DIR=${PIPELINE_DIR}/mojoe
MOJOE_POOL=${PIPELINE_DIR}/testbed/imagepool
MOJOE_TEMP=${PIPELINE_DIR}/testbed/temp
MOJOE_CONFIG=${PIPELINE_DIR}/testbed/config
MOJOE_SERVICE=${PIPELINE_DIR}/testbed/servicelog
MOJOE_DATDIR1=${PIPELINE_DIR}/testbed/datdir1
MOJOE_DATDIR2=${PIPELINE_DIR}/testbed/datdir2
MOJOE_WHATREV=${PIPELINE_DIR}/testbed/Rev
MOJOE_PROTOCOLS_TOP=${PIPELINE_DIR}/testbed
MOJOE_RAW=${PIPELINE_DIR}/testbed/raw
MOJOE_GRAVEYARD=${PIPELINE_DIR}/testbed/graveyard
MOJOE_GATING=${PIPELINE_DIR}/testbed/gating

#
# ONLY CHANGE THESE IF YOU _MUST_ HAVE A DIFFERENT USER/PASS FOR FTP
#
ftpserver=${sshserver}
ftpservername=${sshservername}
ftpuser=${sshuser}
ftppass=${sshpass}
```

The above code is the pertinent configuration settings in `mojoe/settings.ksh`. The first section deals with the SSH (and FTP) user that MoJoe uses to communicate from the scanner to the grid. If SSH and FTP need to be different users, please edit the appropriate section further down. Future versions of MoJoe should make this section unnecessary.

ref_dir, **fmri_dir**, and **bin_dir** are the full path names to the `spm5_pipeline/ref_functions`, `spm5_pipeline/fmri`, and `spm5_pipeline/bin` directories respectively. On our installations, these directories are not all under the `spm5_pipeline` directory.

For example, the reference functions directory may need to be in a read/write accessible location while the fmri and bin directories can be placed in a more restrictive read-only location.

The **billEmail** and **monitorEmail** are email addresses that email should be sent for the person who would keep track of billing and monitoring. These email settings are also affected by the assorted email settings in **spm5_pipeline/WFU/global.site.ksh**. The **billRemoteDir** and **monitorRemoteDir** are remote directories used to store moJoe session information.

*Note: Even if not being used, the **monitorRemoteDir** and **billRemoteDir** must exist.*

The last section, starting with **PIPELINE_DIR** and the assorted **MOJOE_*** settings depend on the location of MoJoe. If MoJoe is being used in a sample, testing, environment, then the **PIPELINE_DIR** should be the location of the **spm5_pipeline** directory. However, on a scanner, MoJoe should be installed in a “moJoe” directory under the home directory of the user and all these settings should be commented out (with a # as the first character on the various lines). If MoJoe has difficulty finding data on the scanner, this section is available to specify the location of the assorted directories.

1.3.9 Optional Transfer of MoJoe to a MRI scanner

To transfer MoJoe to a MRI scanner, first tar (and optionally compress) the moJoe directory. Then move the tar file to the scanner and extract it. On a scanner, moJoe should reside in the **~/moJoe** directory. Edit the settings.ksh file for specific settings of the scanner being used. Sample commands, from the **spm5_pipeline** directory on the computer which MoJoe was compiled:

```
tar -czvf moJoe.tgz moJoe  
transfer moJoe.tgz to the scanner, place in home directory  
cd ~  
tar -xzvf moJoe.tgz
```

1.3.10 Final installation and setup of MoJoe, if MoJoe is to be installed

Note: This series of commands will have to be run from each scanner (computer) on which MoJoe is installed.

MoJoe uses SSH keys in the transfer of data from the scanner to the computing grid. On the scanner (or sample computer), if **~/ssh/id_rsa.pub** already exist, please skip to the third line. In the commands below, replace **sshuser** and **sshserver** with the respective settings from settings.ksh. **SCANNER_NAME** should be the name of the scanner (computer) which these commands are running on.

```
cd ~  
ssh-keygen -t rsa  
choose the default options, with the suggestion of leaving the passphrases blank  
scp .ssh/id_rsa.pub sshuser@sshserver:~/.ssh/id_SCANNER_NAME.pub  
ssh sshserver  
log in as sshuser  
cd ~/.ssh  
cat id_SCANNER_NAME.pub >> authorized_keys
```

While on the scanner, create the directory `/export/home1/mojoe` or the directory specified by `MOJOE_TEMP` in `settings.ksh`.

```
mkdir /export/home1/mojoe
```

Change to the `mojoe` directory and run the `goMoJoeSetup` command. Follow the onscreen instructions.

```
cd mojoe  
./goMoJoeSetup
```

Note: If using on a sample computer, choosing ESE 14 will allow you to the sample pfile script `spm5_pipeline/testbed/newCapture.ksh`. Otherwise, that file will have to be modified for the specified ESE version.

Lastly, `CDRECORD` should be installed on the scanner in order to burn CD's. `Cdrtools` (<http://cdrecord.berlios.de>) is used to burn CD's. As these utilities often require root privileges, a compressed tar of the applicable files has been placed in the `mojoe` directory called **linux_cdrecordsrc.tar.gz**. This file will fill the `mojoe/CDRECORD` directory. From the `mojoe` directory as a root user:

```
tar -xzf linux_cdrecordsrc.tar.gz
```

Note: If the files included in the tar do no work with the system, the original source is available in the `spm5_pipeline` external directory. Please read the documentation included with the package for installation instructions.

1.3.11 Reference Functions

Autoprocessing requires a paradigm reference number assigned to specific reference function file. The reference function is a text based file containing a header (defining the TR, number of images, number of conditions, type of paradigm, desired contrast comparisons and types) and the actual paradigm definition (indicated by begin and end). A detailed paradigm, used for the optional sample study is given on the next two pages. Comments may be placed in the file by prefixing the comment line with a semi-colon.

`#{WFU_PIPELINE}/ref_functions/AMP/DSST_Hard_1801.se_ref`

```

;This is the Hard DSST event-related paradigm (176 images,42 events)
;
;use 1 to indicate baseline (in this case = trials with only 1
;matching probe number)
;0 indicates null images - ie not to go into analysis
;note that baseline should now be considered a
;condition, but null (0) is not
;
;Use the $ to end long lines to continue into next line
;names lists the conditions in numerical order

```

```

paradigm_name      :      DSST_Hard_1801
email             :      bwagner@wfubmc.edu
tr                :      2
conditions        :      4
images           :      176
ER                :      1
fixed            :      0
mixed            :      0
names            :      cue, pro1match, pro2match, pro3match
contrast_names   :      cue_act, pro1_act, pro2_act,$
                  pro3_act, pro1_gtr_pro2, pro1_gtr_pro3,$
                  pro2_gtr_pro3,pro1_gtr_pro23
contrast_types   :      T,T,T,T,T,T,T,T
contrast_values  :      [1,0,0,0],[0,1,0,0],[0,0,1,0],$
                  [0,0,0,1],[0,1,-1,0],[0,1,0,-1],$
                  [0,0,1,-1],[0,1,-0.5,-0.5]

```

```

fixed_effects_paradigm_name :      DSST_Hard_fixedeffects
Fixed_effects_lookup_code  :      1803
dispersion_derivative      :      0
time_derivative            :      1
time_derivative_weight     :      0
dispersion_derivative_weight :      0
processing_type            :      5
high_pass_filter           :      300
low_pass_filter            :      0
auto_correlation           :      1

```

begin

```

;condition appears in the first column
;time (in seconds) appears in the second column
;The comma separator must be used to separate fields
;this format can be used for single event pseudo random experiments
;to check if the format is valid,use the function read_seref
;make sure conditions above match number of conditions below
;note that baseline is now considered a condition
;ONLY USE 0 FOR NULL IMAGES (ie,to be discarded)

```

```

;-----
;blocked paradigm setup (no ER flag)
;-----

```

```

;blocks ,      2,1,2,1,2,1,2,1,2,1
;time      ,      42.9

```

```

;blocks ,      1,2,1,2,0,1,2,1,2,0,1,2,1,2,0,1,2,1,2,0,$
;          1,2,1,2,0,1,2,1,2,0,1,2,1,2,0,1,2,1,2,0
;time      ,      5,5,5,5,20,5,5,5,5,20,5,5,5,5,20,5,5,5,5,20,$
;          5,5,5,5,20,5,5,5,5,20,5,5,5,5,20,5,5,5,5,20

```

```

;-----
;fixed event setup (ER flag and fixed flag)
;-----

;1      ,      1
;isi    ,      14
;2      ,      10
;isi    ,      30

;-----
;variable event setup (ER flag no fixed flag)
;-----

1      ,      1.036,6.923,13.119,19.12,25.085,$
      ,      31.104,37.209,43.179,49.398,55.517,$
      ,      61.407,67.583,73.55,79.532,85.447,$
      ,      91.574,97.472,103.581,109.779,115.755,$
      ,      121.938,128.003,133.858,139.862,145.966,$
      ,      151.983,158.018,164.173,170.18,176.063,$
      ,      182.228,188.317,194.526,200.495,206.698,$
      ,      212.761,218.72,224.796,230.713,236.867,$
      ,      242.866,249,254.975,260.974,266.942,$
      ,      273.043,279.078,285.011,290.888,297.067,$
      ,      303.084,308.969,315

2      ,      2.436,26.485,62.807,104.981,123.338,$
      ,      135.258,159.418,171.58,183.628,189.717,$
      ,      208.098,220.12,238.267,244.266,250.4,$
      ,      274.443,280.478

3      ,      14.519,20.52,38.609,56.917,74.95,$
      ,      92.974,98.872,111.179,129.403,153.383,$
      ,      165.573,201.895,214.161,226.196,256.375,$
      ,      298.467,304.484,316.4

4      ,      8.323,32.504,44.579,50.798,68.983,$
      ,      80.932,86.847,117.155,141.262,147.366,$
      ,      177.463,195.926,232.113,262.374,268.342,$
      ,      286.411,292.288,310.369

;-----
;straight timing setup (any paradigm)
;-----

;0      ,      40
;1      ,      10
;0      ,      31
;1      ,      1
;0      ,      30
;1      ,      1
;0      ,      40
;1      ,      1
;2      ,      50
;1      ,      600

End

```

Once a reference function is written, it must be linked with its paradigm lookup number. This unique floating point id, followed by the full path to the reference function, is located in the tab delimited *WFU_PIPELINE/ref_functions/paradigm_lookup_table*. Like reference functions, comments may be placed in the file by prefixing the line with semi-colons.

Excerpt from

`${WFU_PIPELINE}/ref_functions/paradigm_lookup_table.template`

```

;-----
; POEM
;-----
1800 /home/bwagner/spm5_pipeline/ref_functions/AMP/DSSTblock_1800.se_ref
1801 /home/bwagner/spm5_pipeline/ref_functions/AMP/DSST_Hard_1801.se_ref
1802 /home/bwagner/spm5_pipeline/ref_functions/AMP/DSST_Hard_1802.se_ref
1803 /home/bwagner/spm5_pipeline/ref_functions/AMP/DSST_Hard_fixedeffects.se_ref

```

As the list of paradigm number and associated reference functions can grow to be quite large, and it is essential that the reference function file exist, a simple check function can be run to check for the existence of the reference function file. It does not check the contents of the reference function for validity.

**`cd ${WFU_PIPELINE}/ref_functions`
`./check_paradigm_lookup_table.ksh`**

IMPORTANT NOTE:

Autoprocessing from MoJoe, a directory of dicoms, or a compressed tar of dicoms will only reference these paradigm lookup numbers if dicom field 0019,10e0 is set to a reference number. Otherwise, a paradigm file will need to be placed in the study's root directory. See section **2.2.4 Autoprocessing 4D files from the Command line** for more details.

Running the included sample study:

Place the lines in the above excerpt in the `${WFU_PIPELINE}/ref_functions/paradigm_lookup_table` or if a `paradigm_lookup_table` does not exist, copy the template over. Then see section: **2.1 Using MoJoe**.

1.4 Compatibility

The `spm5_pipeline` has been tested in the following environments:

	WFU's Pipeline Environment	Testbed
OS Version	Solaris 9 Red Hat Enterprise Linux 4 & 5	Gentoo Linux 2008.0
GCC Version	Solaris: 3.4.2 RHEL: 3.4.6 and 4.1.2	4.1.2
Sun Grid Engine	6.0u8	6.2
Matlab Version	7.3.0.298 (Matlab 2006b)	7.3.0.298 (Matlab 2006b)

The `spm5_pipeline` should run on most linux platforms without modification. The pipeline should run on UNIX (Solaris) without modification although MoJoe may not be able to be built against it.

MoJoe currently runs on GE LX MRI Scanners.

1.5 Referencing the software

2. Using the Software

2.1 Using MoJoe

MoJoe can be called from the command line at the scanner console with:

```
cd moJoe  
./goMoJoe
```

MoJoe will provide a list of examinations in the scanner database, and selectors for CD archival and FTP to a remote directory.

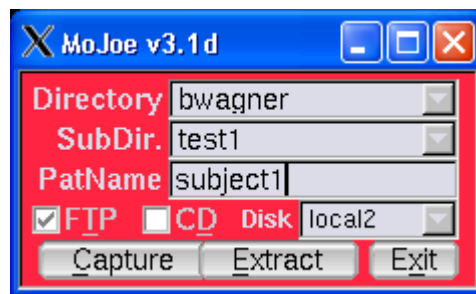


Figure 1. MoJoe. MoJoe startup widget provides pull-down menu for specifying the Directory (generally the Investigator), SubDir (study), and PatName (subject). This information will be stored on the specified Disk. The Capture button initiates fMRI archival monitoring and transfer of RAW p-file data. The Extract button is used for transferring images from the scanner database. FTP archival and CD burning can be turned on or off.

IMPORTANT NOTE:

When capturing, autoprocessing will stop after dicom conversion and 4D image creation if dicom field 0019,10e0 is not set to a valid reference function. The method of setting this field varies by scanner and scanner usage policies.

2.1.1 MoJoe: Capture

When the Capture button is selected MoJoe will automatically monitor for any new raw data files. When a raw data file is detected, it is renamed with the subject label, given a time-date stamp, and ftp'd to the specified disk. The file is then burned to a CD at the scanner console while scanning continues. When the burn is complete, the raw data file is removed from the scanner hard drive and the program continues to monitor for additional raw data acquisitions.

If MoJoe is being run on a test computer, then a sample capture can be done by opening two terminal windows. In one, start moJoe. Initiate the capture routine. Then in the other window:

```
cd ${WFU_PIPELINE}/testbed  
./newCapture.ksh
```


After a few “waiting 15” lines, press Ctrl-C to break the script. MoJoe will then finish the capture.

Note: The p-files included in this sample script are for ESE14. If using a different version, change the last line of the script to the appropriate version.

2.1.2 MoJoe: Extract

When the Extract button is selected a pull down menu and selectors will appear for selecting the series to be transferred. Once selected, the images are extracted from the scanner database, assembled into a compressed tar file, ftp'd to the specified disk, and then burned to CD.

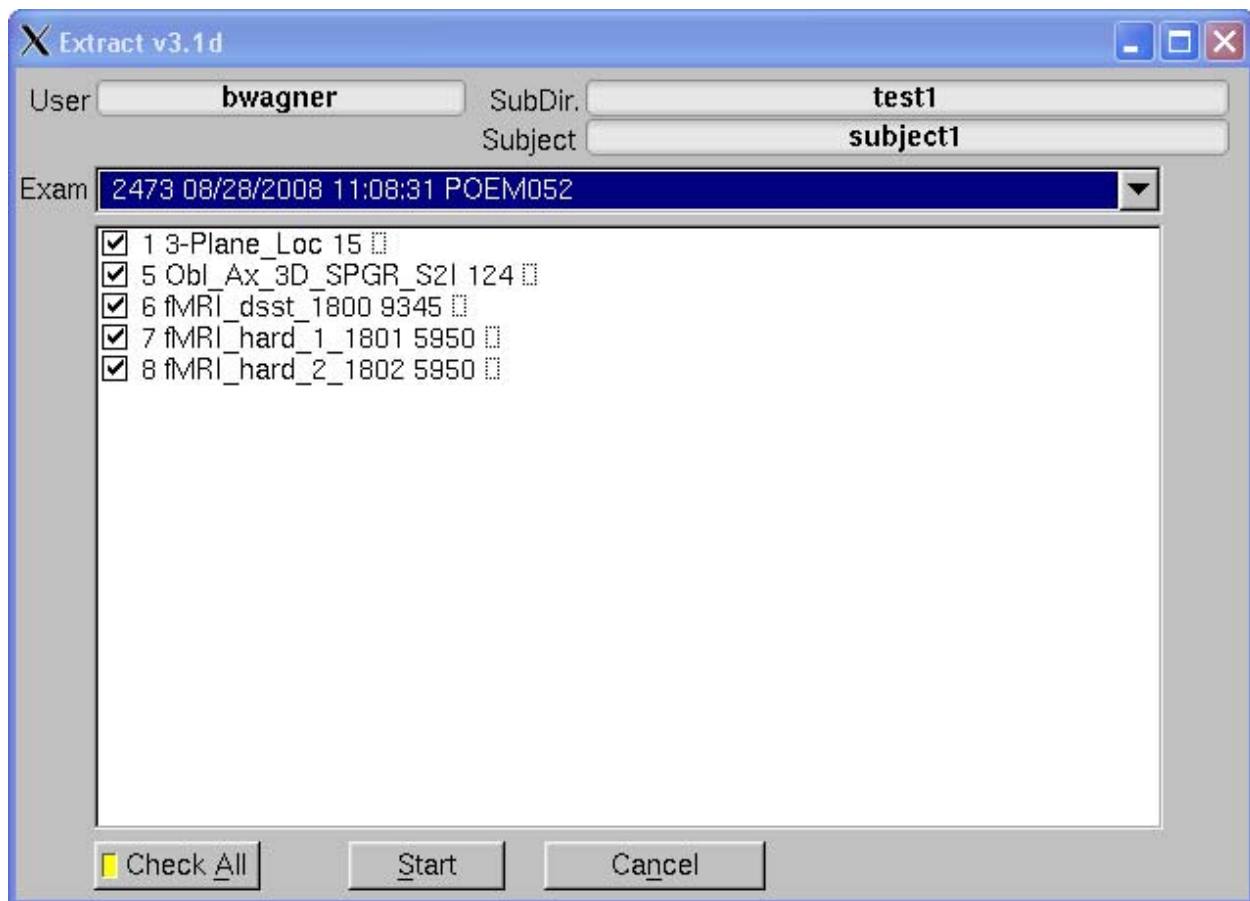


Figure 2. MoJoe Extract Widget. The extract widget provides pull-down menu listing exam number and subject names for data in the scanner database (Exam). Subject label (subject1) is used to label data during transfer and archival. Selected series to be transferred can be checked off in the check boxes within the list of series.

If MoJoe is being run on a test computer, then a sample extract can be done. The above screen should be available. Simply click the start button to start processing.

2.2 Autoprocessing

2.2.1 Autoprocessing from MoJoe

Autoprocessing from MoJoe starts automatically from the completion of a Capture or Extract session. Captured files (p-files) are reconstructed starting by issuing a `${WFU_PIPELINE}/fmri/autorecon_excite` command on the ssh server. Extracted files (dicoms) are processed by issuing a `spm5_pipeline/fmri/autolocsort_excite_paradigm` command on the ssh server. Further details are found in the following sections

2.2.2 Autoprocessing P-files from the Command line

Autoprocessing of p-files is done by the Matlab function `wfu_recon.m`. To simplify processing, a ksh script can be run that will run this Matlab script and set applicable environmental variables:

```
${WFU_PIPELINE}/bin/wfu_RECON [path]pfile1 [path]pfile2 ...
```

IMPORTANT NOTE:

`${WFU_PIPELINE}/WFU/WFU_geToolbox/recon/wfu_recon.m` will need to be modified to suit p-file reconstruction based on specific aims of the study. As each type of reconstruction can vary widely, this script is provided as a starting point for integrating your own p-file reconstruction routines.

2.2.3 Autoprocessing DICOM images from the Command line (*DICOM images that do not have the paradigm number set*)

Processing DICOM images that do not have their paradigm number set (DICOM field 0019,10e0) requires dicom conversion, 4D image creation, creating a file/paradigm pair text file, and then autoprocessing. Given a study directory and within that directory, a directory of dicom images called `dicom_images`, the following command would be called:

```
${WFU_PIPELINE}/bin/wfu_CONVERT_DICOM study_directory dicom_images
```

Create the `processing.paradigms` textfile, see below.

```
${WFU_PIPELINE}/bin/wfu_AUTOSPM5 /full/path/to/study_directory
```

The `wfu_CONVERT_DICOM` command will create several 4D files in the study's root directory. These then must be linked to the appropriate paradigm number in a `.paradigms` file. The name before the `.paradigms` does not matter, but there may only be one `.paradigms` file in the study's root directory. Waiting until after the `wfu_CONVERT_DICOM` step for generation of this file is due to the unique structure of the 4D file names: (subjected-series number-slice number-number of images-PSD name `_4D.img/hdr`). A sample `processing.paradigms` file for the included study is given below.

Sample processing.paradigms

POEM052-0006-267-0001-9345-epiflexfmri_4D.img	1800
POEM052-0007-170-0001-5950-epiflexfmri_4D.img	1801
POEM052-0008-170-0001-5950-epiflexfmri_4D.img	1802

2.2.4 Autoprocessing DICOM images from the Command line (DICOM images that have the paradigm number set)

Autoprocessing of DICOM files can occur in two ways: with either a tar.gz of DICOMs or a directory of DICOM images (under the study's root directory).

`${WFU_PIPELINE}/bin/wfu_FULLPROCESS path/to/study_directory/study.tar.gz`
or
`${WFU_PIPELINE}/bin/wfu_FULLPROCESS path/to/study_directory/dicom_images`

The sample study can be run by typing (as one line):

```
${WFU_PIPELINE}/bin/wfu_FULLPROCESS  
${WFU_PIPELINE}/poem052_08_28_08/poem052_08_28_08_3235_tmp.tar.gz
```

If creating a tar file for processing without the assistance of MoJoe, it is recommended that you follow this form for the tar file structure. The `autolocsort_excite_paradigm` script will expect the archive to be based on a 'tmp' directory. The directory structure afterwards may be based on preference.

Sample Directory Structure of a DICOM tar.gz Archive from MoJoe

```
tmp  
-02473          #exam number  
| -0001        #series number  
| | - I0001.dcm #image  
| | - I0002.dcm #image  
| | - ...etc  
| -0005        #series number  
| | - I0001.dcm #image  
| | - I0002.dcm #image  
| | - ...etc  
| -0006        #series number  
| | - I0001.dcm #image  
| | - I0002.dcm #image  
| | - ...etc
```

Note: Exam, series, and images may be named differently. For example e2473, s1, I01.MRDC.1, respectively.

2.2.4 Autoprocessing 4D files from the Command line

If 4D analyze or nifti data files already exist in a study directory, along with a structural image, then autoprocessing can be completed by:

Creating a processing.paradigms file. See section 2.2.3 Autoprocessing DICOM images from the Command line (DICOM images that do not have the paradigm number set) for details creating this file.

Create a directory in the study root called nifti and place the structural image in this directory.

```
${WFU_PIPELINE}/bin/wfu_AUTOSPM5 /full/path/to/study_directory
```

2.3 Reprocessing and Error Recovery

2.3.1 *cleanrun*

The *spm5_pipeline/bin/cleanrun* command will clean out a subject directory, removing all processing data and assorted file-lists (flists), ques, etc. It will leave the locs, DICOM archive, p-files, and user created directories. Several files will also be deleted from the log directory. If a subject is to be reprocessed, it is recommended that this command be run first. This script will not restart processing.

```
`${WFU_PIPELINE}/bin/cleanrun /path/to/subject_directory
```

2.3.2 *rerun*

The *spm5_pipeline/bin/rerun* command will reprocess a study starting with DICOM conversion. Before reprocessing, several files and directories are deleted then the reprocessing is placed on the grid. This command will work only if data came from a MoJoe extract, manually using *autolocsort_excite_paradigm*, or processing from dicoms. It is recommended to run *cleanrun* before rerunning.

```
`${WFU_PIPELINE}/bin/rerun /path/to/subject_directory
```

2.3.3 *rerun_local*

The *spm5_pipeline/bin/rerun_local* is the same command as above, but the initial DICOM conversions will be run on the computer which the command was issued from. This command will work only if data came from a MoJoe extract, manually using *autolocsort_excite_paradigm*, or processing from dicoms. Again, it is recommended to run *cleanrun* before rerunning.

```
`${WFU_PIPELINE}/bin/rerun_lcoal /path/to/subject_directory
```

2.3.4 *quick_start*

Autoprocessing creates several 'batch_que' and 'batch.flist' files in directories such as the log/batch, batch/RUNx, and FIXED_EFFECTS directories. If processing completes normally, then the batch_que files should all have a file size of 0. However, if an error occurs one or more of these files will still have commands that need to be executed. After the error is fixed, a batch_que can be continued by issuing the *quick_start* command found in the batch_que's directory. To get a listing of all the batch_que's in a subject folder, type:

```
find . -name "batch_que" | xargs ls -lhrt
```

To completely restart a batch_que, copy the batch.flist file over the batch_que file. Then edit the batch_que, removing the number that might be found at the top. Then the *quick_start* command can be issued.

3. Technical Notes

3.1 Transfer of fMRI analysis to PACS

The WFU SPM5 Pipeline will automatically convert an analysis to DICOM and attempt to send to a PACS server if the study's path contains the word *clinical*. The pacs server information is stored in the file

``${WFU_PIPELINE}`/WFU/WFU_dicom5/wfu_dicom_magilla.m` Matlab file. To change the pacs server to which files are sent edit the lines:

``${WFU_PIPELINE}`/WFU/WFU_dicom5/wfu_dicom_magilla.m` excerpt

```
def_flags.send.pacs_host = 'KENNY';  
def_flags.send.pacs_host_ip = '152.11.233.220';  
def_flags.send.pacs_port = 4006;  
def_flags.send.pacs_aetitle = 'FMRI'; %-a flag  
def_flags.send.pacs_dbdir = 'MRAW'; %-c flag
```

To change the default directory of the trigger for automatic conversion to DICOM and sending to pacs, edit: ``${WFU_PIPELINE}`/WFU/WFU_batch/wfu_X_scriptwrite.m`

For a GUI interface to dicom conversion and sending to PACS and additional features, please download our WFU_DICOMtk. This toolbox is for the reverse conversion of Analyze or NIfTI images to DICOM and for inserting the converted DICOM images into a PACS database. This is very useful for clinical implementations in which it is necessary to have post-processed images available for system-wide review. The WFU DICOM toolbox also provides routines for coregistering SPM processed and normalized fMRI t-maps back to native anatomical MRI space (using inverse transformation matrices), and generating fused functional-anatomic images based on user-definable thresholds. If a DICOM header exists for the native space anatomic image, this information is used in converting the fused images to DICOM format. This allows the post-processed images to be linked with the patient's conventional MR data and retrieved as part of the same study on any PACS station during review. The studies maintain all subject identifiers and retain all cross-referencing capabilities. If a DICOM header does not exist for the conversion, a generic DICOM header structure is created, and some user-editable DICOM header fields are provided in the GUI to allow customization (e.g., subject id, study type).

3.2 WFU Grid Distribution Tool

In addition to the non-interactive command line based pipeline, we have also created a generic grid-processing architecture GUI which allows automated distribution of MATLAB programs or scripts.



Figure 3. WFU Grid Distribution Tool. The user can choose the format of the input data (list of filenames, an image, or loop), the distribution partitioning scheme (point, slice, or vector), and the Matlab command to be distributed. All processing information is stored in the Log Directory

This is very useful for programs that need to run on many files or images (for example, preprocessing of MRI images for a VBM analysis on 30 subjects). The only requirement of the MATLAB script to be distributed is that it be structured to take an index of values, a binary image mask of voxels, or list of filenames to operate on as input. Our architecture takes advantage of the Sun Grid Engine “array job” task distribution feature. The grid distribution tool is designed to take any appropriately written MATLAB function, and distribute it on a computing cluster using the Sun Grid Engine array job feature. Further information may be found in: *\${WFU_PIPELINE}/WFU/WFU_grid/README*.

References

Maldjian JA, Baer AH, Kraft RA, Laurienti, PJ, Burdette JH. Fully automated processing of fMRI data in SPM: from MRI scanner to PACS. *Neuroinformatics* 2009.