MMPS Reference Manual

Preface

The Multi-Modal Processing Stream (MMPS) is a software package that consists of binaries, scripts, and matlab functions, designed collectively to process data from non-invasive brain imaging methods. These modalities include sMRI, DTI, fMRI, PET, MEG, and EEG, representing a range of temporal and spatial resolutions, physiological and anatomical sensitivities, and fields of view. The challenge therein is to combine simultaneous and complimentary information from different imaging techniques in order to provide comprehensive analyses across a variety of research applications; the MMPS provides a simple yet powerful interface to do just this, and has aided the understanding of normal function in sleep, memory and language, development and aging, and diseases such as dementia, epilepsy, and autism.

The tools in this package were written by several members of the UCSD Multimodal Imaging Laboratory (MMIL), including: Don Hagler, Anders Dale, Vijay Venkatraman, Dominic Holland, Nate White, Cooper Roddey, Alain Koyama, Jason Sherfey, Rajan Patel, and Ben Cippolini.

Table of Contents

Preface.....2
Table of Contents.....3
Overview.....4
Getting Started.....6
Data Processing.....12
Appendix.....17

Overview

In order to take advantage of the MMPS, a number of conditions must be in place. First, a user account is created, a project is defined, and a data storage partition is designated. A fixed directory structure is established within the account, and a few essential files are customized for the user and the project. The *MMIL Server Administrator* will help set up new accounts and projects; additional information regarding these steps is available in the section Getting Started.

With those formalities in place, new data can be transferred to the designated server location. There are a few ways to transfer data to the MMIL servers (details here), and generally the initial input for the MMPS is raw data files directly from the imaging scanner. Depending on the size and scope of the project, the data may be processed by employing the standard stream of the MMPS or by using individual, customizable scripts. Both methods provide the same result, but the steps required by the user are slightly different.

The general workflow of the MMPS is as follows:

1. Process raw data
2. Inspect raw and processed data for errors
3. Perform detailed editing and reprocessing, if necessary
4. Compute averages and group analysis

More specifically, for each modality the following processes and transformations are applied:
Structural MRI: Raw MRI data is unpacked, classified, converted from DICOM to .mgh/.mgz file format, gradient and B0 unwarped, and resampled. T1 images are rigid-body registered. Using FreeSurfer, a cortical reconstruction is created.
DTI: Diffusion-weighted images are corrected for eddy current distortion, head motion, and B0 distortion.
BOLD: Images are corrected using the reversing gradient method.
MEG: Forward solutions are calculated using the boundary element method (BEM), and MEG

The user account *mmilrec19* exists for purposes of training. This account is set up to run the MMPS and has examples of MEG and MRI data, and is referred to throughout this guide. You may view and copy files from this account to your own, or you may request access if you do not yet have an account.

Recharge Projects

Recharge projects are defined as metered services (‘pay-per-use’) for affiliated Investigators who process imaging data using the MMPS and MMIL resources. Most projects are large-scale, longitudinal studies with hundreds (if not thousands) of imaging sessions. These data are processed under the *REC* version of the MMPS by a designated

team of MMIL staff members. Recharge projects are strictly managed with the goal of providing high-quality, uniform, and reliable results to project PI's. By contrast, non-recharge projects do not pay for data processing, are generally small-scale, and are managed by local Investigators at the MMIL. These projects may utilize the standard MMPS pipeline or a small set of defined scripts that source the MMPS, and data quality standards are unique to the study

While recharge projects follow a regimented method and data is processed under *mmilrec* user accounts, non-recharge projects may be processed using personal user accounts (though data resides on another partition).

MMIL Compute Clusters

The MMIL has a number of networked clusters available for data processing. Jobs are created by running particular scripts.

FreeSurfer

“Cortical Surface reconstruction”

Output and Analysis

The output of the MMPS is...
ROI Summaries: .csv files with ROI areas, cortical thickness
May import summaries for analysis with matlab, SPSS, MS/Open Office
Concatenate surfstats: compiles all subject data into one 4D file
analysis must be biologically-based, that is, the variables extracted need to reflect biological rather than statistical phenomena;

Understanding This Guide

- Examples of terminal inputs are indented and designated by ‘$’ for c-shell commands and ‘>’ for MATLAB commands.
- Terms that are boldfaced and italicized (e.g. ***username***, ***server_partition***, ***Proj_ID***, ***SubjID***) are placeholders for your unique username, project name, data server partition, etc.
- The wildcard symbol ‘*’ is used to abbreviate and include all files or directories in a particular location; for example, *MRIRAWDICOM** refers to one and all subject directories within the MRI/raw/ directory.
- The term ‘ip#’ is a placeholder for your unique IP address, and may be expressed as an IP (137.110.172.53) or as a hostname (ip53.ucsd.edu).
- The user account ***mmilrec19*** may be used in examples, as this account and its contents are available for training purposes.

Getting Started

Create a User Account

For all new user accounts, contact the *MMIL System Administrator*, who will provide you with a *Space Request Form* – fill out this form and return it. This step should be done as far in advanced as possible to allow time for space to be made available. You will then be issued a new user account and a designated NFS server partition in which to store your data. The file path of your data partition will look something like this:

```
/space/md16/1/data/MMILDB/Proj_ID/
```

Do ***not*** permanently store data in the */home* partition (where your base user account resides) because it will not be properly backed-up and filling this partition to capacity will cause problems for all MMIL users.

Create a Project

Refer to the sections below to create data directories and associated project-specific files.

Data Directories

Within the server partition designated to you in the previous step, data directories must be created. Navigate to the server directory:

```
$ cd /server_partition
```

Paste the following into the command prompt:

```
set MEG_dirlist = (orig raw proc)
set MRI_dirlist = (orig raw proc fsurf fsico)
foreach MEG_dir ($MEG_dirlist)
  mkdir 'MEG/'$MEG_dir
end
foreach MRI_dir ($MRI_dirlist)
  mkdir 'MRI/'$MRI_dir
end
```

This sequence of commands will create two directories, *MEG* and *MRI*, and within each are the respective sub-directories as indicated in their *dirlist* above. You may also create these directories manually using the *mkdir* command. List each folder to verify its contents:

```
$ ls MEG
orig   proc   raw

$ ls MRI
fsico   fsurf   orig   proc   raw
```

Also, create a *data* link within your home directory for easy access to the NFS location, and use the project name (the same project ID as in your *MMIL_ProjInfo.csv* file) in the link name:

```
$ ln -s /server_partition /home/username/data_Proj_ID
```

The default directory structure for your project is now in place. List the new link to verify its contents:

```
$ ls /home/username
Desktop   MetaData   ProjInfo   batchdirs   data_Proj_ID   matlab
```

```
$ ls /home/username/data_Proj_ID
MRI   MEG
```

Project Info File

Copy the *MMIL_ProfInfo.csv* file from the *mmilrec19* account to your own account:

```
$ cp /home/mmilrec19/ProjInfo/MMIL_ProjInfo.csv /home/username/ProjInfo/
```

Open the file with Open Office:

```
$ ooffice /home/username/ProjInfo/MMIL_ProjInfo.csv
```

Other than the header row, each successive row represents a project. For the first project, in row two, modify the 'ProjID' field to use a unique project name (***Proj_ID***, the same name used for the data directory link above), and the 'PI' to the study's primary investigator. The next eight columns specify the location of data directories related to this project; edit these values using the file paths for the data directories that you created in the previous step. The values for the remaining columns are standard and may be left as is. More information about this file is available in the Appendix under Directory Structure.

Visit Info File

Within the *ProjInfo* directory, create a directory using the project name listed in the *MMIL_ProjInfo.csv* file:

```
$ cd /home/username/ProjInfo
$ mkdir Proj_ID
$ ls /home/username/ProjInfo
MMIL_ProjInfo.csv   Proj_ID
```

Copy the ***Proj_ID***_*VisitInfo.csv* file from the *mmilrec19* account to your own account:

```
$ cp /home/mmilrec19/ProjInfo/TST_dSPM/TST_dSPM_VisitInfo.csv \
  /home/username/ProjInfo/Proj_ID
```

Rename the file using the project name:

```
$ cd /home/username/ProjInfo/Proj_ID
$ mv TST_dSPM_VisitInfo.csv Proj_ID _VisitInfo.csv
```

Open the file with Open Office:

```
$ ooffice /home/username/ProjInfo/Proj_ID/Proj_ID_VisitInfo.csv
```

Other than the header row, each successive row represents a single subject. For the first subject, in row two, modify the 'SubjID' field to use a unique subject name. For the 'MEG_VisitID' and 'STRUCT_VisitID' fields, list the respective names of the directories containing the raw MEG and MRI data files (that you will transfer to the *orig* data directories in the Copy Data to the Server step). The value for 'VisitID' field is usually the same as the 'STRUCT_VisitID' field, but it may be different; Note that in this file you can list the same subject ('Subj_ID') in multiple rows, each using different MEG and MRI datasets (or visits). More information about this file is available in the Appendix under Directory Structure.

Other File Configurations

For recharge projects, these configurations should already be in place.

cshrc

The *.cshrc* file exists in your home directory (*/home/**username** aka ~/*) and is a C-shell startup configuration file that specifies software environments, file paths, shell variables, and aliases. This file is usually created during the process of establishing a user name, but if it does not exist, copy the *.cshrc* file from the *mmilrec19* user account to your account:

```
$ cp /home/mmilrec19/.cshrc /home/username/

or

$ cp /home/mmilrec19/.cshrc ~/
```

Open the file with a text editor (nedit):

```
$ nedit /home/username/.cshrc

or

$ nedit ~/.cshrc
```

Note that under the section '*Setup standard software environment*' that a particular version number of the MMPS is sourced:

```
$ source $PUBSH/bin/SetUpMMPS.csh 225
```

To determine which version of the MMPS to use, ask your supervisor, or check the following directory for the most recent stable release:

```
$ ls /usr/pubsw/packages/MMPS
```

The most recent version will be listed within this directory as 'MMPS_225' or similar, with higher numbers being later releases; edit your *.cshrc* to use this version unless otherwise specified by your supervisor.

Likewise, certain versions of MATLAB and FreeSurfer are sourced in the *.cshrc*. Consult your supervisor to determine which versions to use.

Aliases are user-defined shortcut commands and may be listed in the *.cshrc* file. To see a list of aliases commonly-used at the MMIL, see Aliases.

MATLAB Startup

The *startup.m* file exists in */home/**username**/matlab* and specifies user-defined options for use in MATLAB. For the MMPS to run, the following line needs to be in the *startup.m* file:

```
run('/usr/pubsw/packages/MMPS/startup_MMPS');
```

Other options and paths may also be added to the file. In order to use the optional MMPS Custom Scripts, copy the script directory to your matlab directory, and add the following line to your *startup.m* file:

```
addpath('/home/username/matlab/MMPS_scripts');

or

addpath('~/matlab/MMPS_scripts');
```

Copy Data to the Server

Copy individual imaging visits to the *orig* data directory. The MMPS will process new data from this directory location. To determine a partition's available disk space, type at the command line:

```
$ df –h
```

Only partitions that have been navigated or linked to are visible, and other projects may also require some of the available space. There are multiple methods of data transfer listed below.

Secure Copy

From a remote computer with access to secure shell, data may be copied to the MMIL servers via secure copy. Only a select number of MMIL Linux workstations allow outside access. From within the local directory containing your data directory, type at the command line:

```
$ scp –r examdirectory username@ip#.ucsd.edu:/home/username/data_Proj_ID/orig
```

Enter the user password, and accept keychain access if necessary. Log out when finished.

Hard Drive

If you have an external USB hard drive (formatted as FAT32) that contains your data, plug it in to a MMIL Linux computer. The drive will be automatically recognized and will appear on the desktop and at the directory location:

```
/media/drivename
```

Copy the data from the drive to your *orig* data directory on the MMIL server:

```
$ cp –rp /media/drivename/* /home/username/data_Proj_ID/orig

or

$ cp –rp /media/drivename/* ~/data_Proj_ID/orig
```

This command copies all contents of the hard drive to the designated location on the

server; use an appropriate file path if your data resides in a sub-directory on the hard drive.

OsiriX

Data may also be transferred to the MMIL from off-site locations via PACS transfer to the local OsiriX host. To learn more about OsiriX, visit their website (www.osirix-viewer.com). If your data resides in our OsiriX database, follow these instructions to copy the data to the MMIL server.

1. On the *pacsmmil* iMac, create a folder on the desktop and title it 'Scan'.
2. Open OsiriX from the icon dock at the bottom of the screen.
3. Locate and export the exam from the OsiriX database to the Desktop.
    a. In OsiriX, highlight the desired scans.
    b. Click the *Export* icon at the top of the window.
    c. Select the *Scan* folder on the desktop as the destination.

    If individual scan directories need to be renamed, do so in the *Scan* folder.
4. Open an *X11* terminal from the icon dock at the bottom of the screen, and type at the command line:

```
$ cd /Users/pacsmmil/Desktop/Scan
$ scp –r username@ip#:/home/username/data_Proj_ID/orig
```

5. Enter user password and allow to transfer finish. Log out, and delete the *Scan* folder containing your data from the local desktop.

Data Processing

MMPS Pipeline

Follow the steps below to use the default MMPS to process MRI data.

1. **MMIL_Process_Exams**
2. **MMIL_Freesurfer_Recon_Exams**
3. **MMIL_IcoResamp_FSRecon_Exams**
4. **MMIL_Analyze_Exams**
5. **MMIL_Summarize_MRI_Analysis**
6. **MMIL_Concat_SurfStats**

1. Process Data

```
> MMIL_Process_Exams('Proj_ID')
```

If MRI dataset contains DTI scans, wait for this step to complete and rerun with the following flag:

```
> MMIL_Process_Exams('Proj_ID', 'DTI_ATLflag',1)
```

2. Freesurfer Recon

```
> MMIL_Freesurfer_Recon_Exams('Proj_ID')
```

If edits are made to the FreeSurfer recons, rerun the command to remake the brain surfaces. To help understand what type of editing may be he

3. Resample to Average

```
> MMIL_IcoResamp_FSRecon_Exams('Proj_ID')
```

4. Analyze MRIStats

```
> MMIL_Analyze_MRI_Exams('Proj_ID')
```

5. Summarize

```
> MMIL_Summarize_MRI_Analysis('Proj_ID')
```

6. Concatenate SurfStats

```
> MMIL_Concat_SurfStats('Proj_ID')
```

7. Longitudinal Analysis (optional)

```
> MMIL_Long_Setup_Exams('Proj_ID')
> MMIL_Long_Register_Exams('Proj_ID')
> MMIL_Long_Analyze_Long_Exams('Proj_ID')
```

Note: This step assumes that a *LONGContainers* directory has been created within the MRI data directory.

MMPS Custom Scripts

The scripts used in the previous section rely on the *ProjInfo* file and flags for settings (*ProjInfo* overrides default MMPS settings), but the scripts described below may contain flags and settings hard-coded within. For the user, there's less information to remember; optionally, the user may supply parameters in the command line (command line overrides *ProjInfo* settings).

The user account *mmilrec19* has these MATLAB scripts and a sample dataset available for training purposes under the project name '*TST_dSPM*'. You may practice with this existing data (request the user password), or you may copy these scripts to process your own data. These instructions assume that you have some limited understanding of MATLAB code, and that your user account and project are set up according to the instructions above (see Getting Started).

The matlab scripts are found within */home/mmilrec19/matlab/MMPS_scripts_clean*, and include:

run_proc_MRI.m	run_fsrecon.m
run_proc_MEG.m	run_icoresamp.m
run_dSPM.m	run_createbem.m

To process data under your own user account, copy the *MMPS_scripts* directory from *mmilrec19* to your own ~/matlab directory and edit the name of the project and other parameters as needed.

```
$ cp –rp /home/mmilrec19/matlab/MMPS_scripts_clean /home/username/matlab/

or

$ cp –rp /home/mmilrec19/matlab/MMPS_scripts_clean ~/matlab/
```

Also, ensure that the file path to these scripts is in your *startup.m* file in your ~/matlab

directory. If the following line is not in the *startup.m* file, then add it (replace ***username*** with your own username):

```
addpath(genpath('/home/username/matlab/MMPS_scripts'));

or

addpath(genpath('~/matlab/MMPS_scripts'));
```

General Steps:

1. **Process MRI data**
2. **Process MEG data**
3. **QC of raw data**
4. **FreeSurfer recon**
5. **Resample to average**
6. **Creat BEM surfaces**
7. **Register MEG to MRI data**
8. **Run dSPM**
9. **Make movies (optional)**

1. Process MRI Data

To generate jobs for processing, run the following command at the matlab terminal:

```
> run_proc_MRI
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing.

2. Process MEG Data

To generate jobs for processing, run the following command at the matlab terminal:

```
> run_proc_MEG
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing

3. QC of Raw Data

After the processing jobs have finished, open matlab, navigate to the subject's MEG *proc* container, and then run:

```
> MEG_MMIL_Browse_Raw
```

Configure the browser window:

- Check ‘Show Chan Labels’
- Uncheck ‘EEG’, ‘Other’, and ‘Filter Data’ to limit view to ‘Mag’ and ‘Grad’ channels.
- Change ‘Scaling’ to 1
- Increase ‘disp # chans’ to 50 or 100

Note noisy channels in raw data, particularly ones that cause false rejections. Ensure that blinks in EOG are properly rejected.

If you wish to exclude certain bad or noisy channels, create a *badchans.txt* file within the proc container and list bad channels:

```
MEG0321
MEG1242
MEG2412
```

For an example, see:

```
/space/md5/2/data/hagler/MEG/proc/contrast/MEGPROC_DJH_B001_090811/badchans.txt
```

If any channels are added to the *badchans.txt* file, rerun *run_proc_MEG* to effect these changes as described in step 4 above.

4. FreeSurfer Recon

After the MRI processing jobs have finished (from step 4), process the MRI data through FreeSurfer.

In matlab, run the following scripts to generate jobs for processing:

```
> run_fsrecon
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing.

Note: If jobs are run on a cluster, rerun *xcshjobs* locally to complete cortical ribbon, wmparc, and aparc+aseg. (Only for FreeSurfer v4.x)

Freesurfer recons may require manual editing if the cortical and pial surfaces do not accurately reflect the anatomical borders. We can modify the location of the surface by adding and removing voxels and control points from within the *brainmask.mgz* and *wm.mgz* volumes; at times, this may not have an effect at all. For editing guidelines, refer to the manual in the following NFS location:

```
/usr/pub/packages/MMPS/documentationFreesurfer_Processing_Instructions.pdf
```

Note that after manual edits are made, you musct create the necessary touch file and rerun this step in order to apply the changes. When complete, check the recon again to ensure the surfaces are corrected.

5. Resample to Average

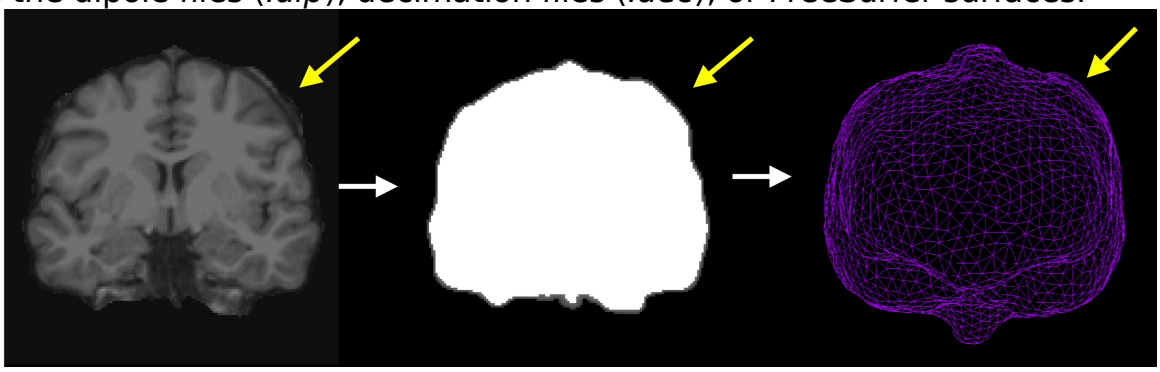
Run the following command at the matlab terminal:

```
> run_icoresamp
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing.

6. Create BEM Surfaces

Boundary Element Methods, or BEM, are required to solve the 'forward problem' by calculating the surface magnetic field given the measured sources inside the head. Because the magnetic field is not distorted by brain tissue, skull, or scalp, only the inner skull surface needs to be calculated. This information, saved as a *.tri* file, is then used to constrain the source solution within the structural MRI *.mgh* file. However, BEM is not used to create the dipole files (*.dip*), decimation files (*.dec*), or FreeSurfer surfaces.



In fact, BEM surfaces rely on the FreeSurfer cortical reconstruction. As such, it is important to ensure the FreeSurfer surfaces accurately represent the anatomical boundaries. For example, the inclusion of dura matter within the pial surface results in an incorrectly-shaped inner skull surface:

The artifact, denoted by the yellow arrow, proliferates through BEM steps from the T1 MRI image to the mask and ultimately to the inner skull surface. To correct for these artifacts, edit the FreeSurfer MRI volumes according to ***these directions***.
Run the following command at the MATLAB terminal:

```
> run_createbem
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing.

7. Register MEG to MRI

In matlab, navigate to the subject's MEG *proc* container, then run the following command:

```
> MEG_MMIL_Reg2MRI
```

Align the HPI points with the surface outer scalp. Click *Save* to accept registration. See:

```
/space/md5/2/data/hagler/MEG/proc/contrast/MEGPROC_DJH_B001_090811/mri2head.trans
```

8. Run dSPM

Run the following commands at the matlab terminal:

```
> run_dSPM
```

Then submit jobs to the cluster or run them locally. For directions, refer to MMIL Compute Clusters and Local Processing.

9. Make Movies (optional)

See...

```
/space/md5/2/data/dhagler/extra/test/test_crosstalk_SNR/run_movies_crosstalk.m
```

and sample_movie_*.m in:

```
/space/md10/8/svn/mmildev/mmildev/mmil/MMPS/matlab/timesurfer/samples
```

Appendix

Common Linux Commands

Linux Manual (man) Pages

```
[testuser@ip70 ~]$ man cp
```

...results in the display of the Linux manual for the **cp** command. The first 'page' of this

manual is shown below...

```
CP(1)                           User Commands                          CP(1)


NAME


       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --archive
              same as -dR –preserve=all

       --backup[=CONTROL]
              make a backup of each existing destination file

       -b     like --backup but does not accept an argument
:
```

Navigate Directories

```
[testuser@ip70 ~]$ cd foldername
[testuser@ip70 pubsw] $ cd ..
[testuser@ip70 usr] $ cd ~
[testuser@ip70 ~]$
```

List Files and/or Directories

```
[testuser@ip70 ~]$ ls
batchdirs      Desktop    matlab_crash_dump.9941  RCS         Templates
bio            Documents  Music                   sample.tgz  testing
data_REC_TEST  Downloads  Pictures                scripts     Videos
data_TEST      matlab     Public                  sge


[testuser@ip70 ~/matlab]$ ls -al
total 24
drwxrwxr-x  2 testuser testuser 4096 Apr  5  2011 .
drwxr-xr-x 51 testuser testuser 4096 Jan 19 14:12 ..
-rw-rw-r--  1 testuser testuser  598 Apr  5  2011 startup.m


[testuser@ip70 ~/batchdirs]$ ls -ltr
total 24
drwxrwxr-x 2 testuser testuser 4096 Jun  2  2010 testcshjobs
drwxrwxr-x 3 testuser testuser 4096 Jun 21  2010 REC_MMIL_Process_Exams_TEST
drwxrwxr-x 3 testuser testuser 4096 Jul 28  2010 testmatjobs
```

Print Working Directory

```
[testuser@ip70 ~]$ pwd
/home/testuser
```

Create Directories

```
[testuser@ip70 ~]$ mkdir foo


[testuser@ip70 ~]$ mkdir -p foo/bar/baz


[testuser@ip70 ~]$ ls -ld ./foo
drwxrwxr-x 2 testuser testuser 4096 Mar 29 11:40 ./foo
[testuser@ip70 ~]$ tree -p foo
foo
`-- [drwxrwxr-x]  bar
    `-- [drwxrwxr-x]  baz

2 directories, 0 files
```

Copy Files and/or Directories

```
[testuser@ip70 ~]$ touch ./bar/baz/foo2
[testuser@ip70 ~]$ cp ~/bar/baz/foo2 ~/foo/bar/baz/
[testuser@ip70 ~]$ tree -p foo
foo
`-- [drwxrwxr-x]  bar
    `-- [drwxrwxr-x]  baz
        `-- [-rw-rw-r--]  foo2


2 directories, 1 file


[testuser@ip70 ~]$ cd ~/bar


[testuser@ip70 ~/bar]$ cp -r ~/foo .


[testuser@ip70 ~/bar]$ tree -p ~/bar


/home/testuser/bar
|-- [drwxrwxr-x]  baz
|   `-- [-rw-rw-r--]  foo2
`-- [drwxrwxr-x]  foo
    `-- [drwxrwxr-x]  bar
        `-- [drwxrwxr-x]  baz
            `-- [-rw-rw-r--]  foo2

4 directories, 2 files


$ cp –p
```

Remove Files and/or Directories

```
$ rm                $ rm –r
```

Common Text Editors

```
$ nedit              $ gedit
```

Command Line Text Processing

```
cat (print entire contents of a file in terminal window)
tail (print last X lines of a file in terminal window)
grep (search for a term in a file/directory)
```

Disk Usage

```
$ du –sh             $ df –h

NOTE: only partitions that have been navigated or linked to are visible.
```

Symbolic Links

```
$ ln –s target link-name
```

Command Help Files

Most commands used at the terminal prompt (including many of the common commands listed in the previous section) contain a help file with a description of usage and optional parameters. For example, to view the help file for the command *mri_info*, type:

```
$ mri_info –help
```

Similarly, MATLAB commands also contain help files that describe required input and optional parameters. For example, to view the help file for the command *MMIL_Process_Exams*, type:

```
> MMIL_Process_Exams
```

.cshrc (c-shark)

The *.cshrc* file exists in your home directory (*/home/**username***) and is a C-shell startup configuration file that specifies software environments, file paths, shell variables, and aliases.

To view or edit the current user's *.cshrc* file, use one of the following commands:

```
$ nedit ~/.cshrc
$ gedit ~/.cshrc
```

Software Environments and Packages

Aliases

Aliases are user-defined shortcut commands, and are a way to simplify the execution of long command strings. They are defined in either the *.cshrc* or *.alias* file in the base directory of a user account (*/home/**username***). An unlimited number of aliases may be created. Listed below are aliases commonly used within the MMIL network.

MMIL Compute Clusters

```
$ alias smc ‘ssh –X mmilcluster’
$ alias smc3 ‘ssh –X mmilcluster3’
$ alias smc4 ‘ssh –X mmilcluster4’
$ alias qstatall ‘qstat –u “*”
```

FreeSurfer Editing and QC

```
alias sete  'setenv SUBJECTS_DIR ${PWD}'
alias tke 'tkmedit \!:1 brainmask.mgz lh.white -aux T1.mgz -aux-surface rh.white -tcl
/space/md8/6/data/MMILDB/RECHARGE/HAW_METH/FSContainers/bm.tcl'
alias tkwm 'tkmedit \!:1 wm.mgz lh.white -aux brainmask.mgz -aux-surface rh.white -tcl
/space/md8/6/data/MMILDB/RECHARGE/HAW_METH/FSContainers/wm.tcl'
alias tka 'tkmedit \!:1 orig.mgz -aux T1.mgz  -segmentation aseg.mgz
$FREESURFER_HOME/FreeSurferColorLUT.txt -tcl /home/mmilrec/bin/tkmedit_aseg.tcl'
```

For MRI DTI QC

```
alias tkregUW 'tkmedit -f \!:1/DTI3_B0uw_f0.mgz -aux \!:1/DTI2_rev_B0uw_f0.mgz'
alias tkregREG 'tkmedit -f \!:1/FLASHhi1.mgh -overlay \!:1/DTI3_corr_f0_resT1.mgh'
alias tkregFA 'tkmedit -f \!:1/T1_resDTI.mgh -aux \!:1/DTcalc/*FA.mgz'
```

MATLAB

/home/username/matlab/startup.m

The *startup.m* file exists in */home/**username**/matlab* and specifies user-defined options for use in MATLAB.

/home/username/.cshrc

To view or edit the current user’s *.cshrc* file, use one of the following commands:

```
$ nedit ~/.cshrc
$ gedit ~/.cshrc
```

Terminal or GUI

Directory Structure

At the base level of your user account (/home/***username***), there are a number of essential files and directories. The integrity of this file structure is necessary for the MMPS stream to function properly.

/home/*username*/*data_Proj_ID*

Note: Replace '*Proj_ID*' with your current working project name, as designated in your project info file, *MMIL_ProjInfo.csv* (see /home/username/ProjInfo).

Most likely, your data will be located on another partition, and those data directories are linked here with the following sub-directory names:

- ***orig*** – New visits transferred to this folder will be processed by the stream.
- ***raw*** – Unpacked data from *tmp* directory into *RAWDICOM** folders for each visit. The *SeriesInfo.csv* file contains info about scans, type of series, number of images per scan, etc.
- ***proc*** – Processed data from *raw* directory into *MRIPROCESSED** folders for each visit. Contains scans in mgh format (registered to atlas space and unwarped); also contains mat file that transforms the scan into subject space.
- ***fsurf*** – Processed data from *proc* directory into *FREESURFERRECON** folders for each visit.
- ***fsico*** – Scans have been resampled to an average brain. This dataset is used for statistical analysis, mostly used for the average brain.

/home/*username*/ProjInfo

The file, *MMIL_ProjInfo.csv*, defines key variables required by the MMPS to correctly process image data. This includes a project name, location of data directories, type of data, FreeSurfer version and recon flag for MRI data, and an array of specific parameters for MEG data. Multiple projects may exist, each with unique settings. When running the MMPS on particular set of data, you will need to know the corresponding project name (***Proj_ID***).

Also here are directories titled with each project name (***Proj_ID***) identified in *MMIL_ProjInfo.csv*. Within each of those directories is a file, ***Proj_ID***_*VisitInfo.csv*, that specifies a list of subjects (or visits), names of their MEG and MRI data directories, and a QC value for their data. The '*QC*' (quality control) field is essentially used to include or exclude certain subjects from processing and analysis.

For recharge projects, this directory and its contents are owned by *mmilrec* and can be viewed but not be edited by you. Please request any changes from the *mmilrec administrator*.

/home/*username*/batchdirs

Each folder name within this directory is project-specific (e.g. *MMIL_Process_Exams_**Proj_ID***), based on the project name and script used to create the batch of jobs. Within each folder are the following files:

- *job_001_**SubjID**.m* or *job_001_**SubjID**.csh* – Contains MATLAB or csh job that will be run once the project is put on the cluster (one job for every subject)
- *scriptlist.txt* – Contains a list of jobs that will be run once the project is submitted to the cluster. If you do not want all jobs to be run, edit this file to include only the desired jobs.
- ***/pbsout*** subdirectory **–** Contains the following output files after the respective jobs have been run:
    - *job_001_**SubjID**.out* –Printed output of the entire job process, including information about any errors or failures that have occurred.
    - *job_001_**SubjID**.err* – Error output if an error is encountered during processing; this file is usually empty if processing completed successfully and correctly.

/home/*username*/MetaData/*Proj_ID*

This directory contains various files that are used throughout the recharge processing pipeline:

- ***Proj_ID**_DTIRegQC.mat* – Created when using *MMIL_Viewer* to perform registration QC. Contains rating scale and notes for each visit.
- ***Proj_ID**_RawQC.mat* – Created when using *MMIL_Viewer* to do RawQC. Contains rating scale and notes for each visit.
- ***Proj_ID**_VisitInfo.csv* – Contains Subject ID, Scan ID, and Visit Number for every processed scan (Created and updated manually).
- *REC_**Proj_ID**_AsegQC.csv* – Created manually during ASEG QC, denotes which areas are over or underestimated, and which should be rejected.
- *REC_**Proj_ID**_dvQC.csv* – Created manually when doing dv QC.
- *REC_**Proj_ID**_FSReconQC.csv* – Created manually during Freesurfer Surface Edits.
- *REC_**Proj_ID**_PET_QC.csv* – Created manually when doing PET registration.

/home/*username*/MetaData/*Proj_ID*/ROI_Summaries/

This directory contains metrics, once Freesurfer recons, PET, or DTI have been analyzed and summarized.

MMPS Functions and Options

Upgrading to new MMPS version

If you are upgrading to a newer version of the MMPS and have existing data that you've processed with earlier versions, it is best to delete the *raw* and *proc* containers and reprocess the data from the *orig* container. This is not strictly necessary, depending on

what previous version was used and what types of data you have, so feel free to inquire about your particular circumstances.

MMIL Compute Clusters

The MMIL has three clusters running SGE available for data processing. Their hostnames are *mmilcluster, mmilcluster3, and mmilcluster4*.

Logging In

From any Linux computer on the MMIL network, use a c-shell terminal to log into a cluster. For instance, to log into *mmilcluster4*, type at the command line:

```
$ ssh –X mmilcluster4
```

Or, if the common alias command is designated in your ~/.cshrc (refer to .cshrc), type:

```
$ smc4
```

Keep in mind that you will be logged into the cluster until you log out. Log in only to submit jobs, check their status (details in next section), and promptly exit when finished. To log out of the cluster, type:

```
$ exit
```

Frontend Processing

The host that you log in to when working on MMIL compute cluster is called the 'frontend' node. This machine is the brains of the compute cluster and is responsible for managing cluster resources and jobs, for providing access to the cluster to users, and for translating file I/O requests between compute nodes and NFS servers providing MMIL /space/* directories. If the frontend node becomes 'loaded' with user applications/processing then access to the cluster can be compromised. As such, performing analyses on the frontend node is **NOT** permitted at any time. **NEVER** run user applications such as MATLAB, R or FreeSurfer on the cluster frontend.

Batch vs. Interactive Jobs

The job scheduler in MMIL compute clusters is called Sun Grid Engine (SGE). SGE manages jobs submitted to the compute cluster allocating jobs to compute nodes based on resource requests and availability.

Most jobs used with the MMPS workflow are termed batch jobs and they run unattended when resources are available. SGE and our cluster software also support the running of interactive jobs.

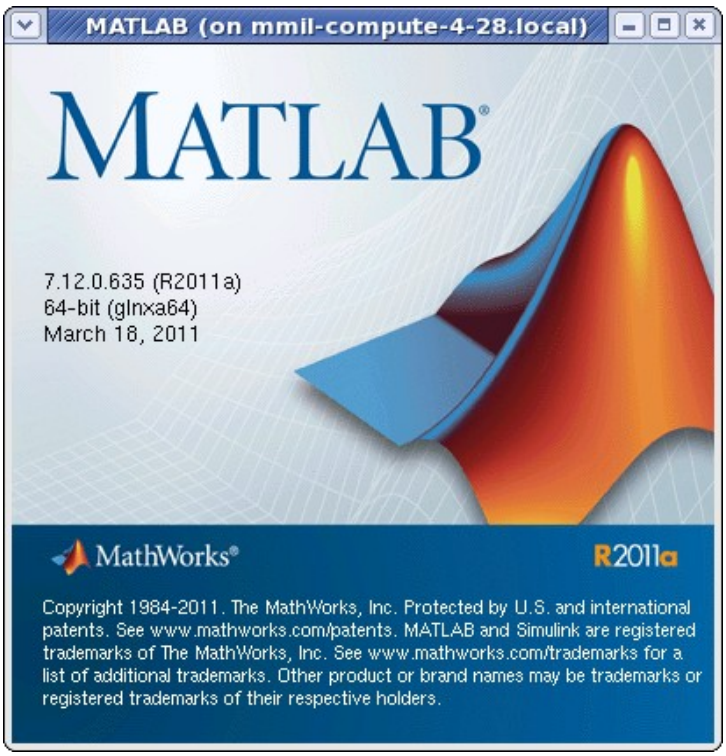
Essentially, an interactive job is one that is scheduled to run on a compute node with available resources and is 'typically' just a login shell. From that shell the user running the job can interactively process data on the compute cluster. Effectively any single job that can be run as a batch job can be run in an interactive session. In addition, jobs requiring user input during the job can also be run. Input can be supplied at the command line or in a graphical user interface.

Starting an interactive job on the MMIL clusters is very easy. Simply execute the following commands in sequence...

```
[username@ipXX]$ ssh -X mmilcluster4.ucsd.edu
Last login: Tue Mar 20 17:20:12 2012 from ip70.ucsd.edu

Rocks 5.4 (Maverick)
Profile built 23:54 15-Jul-2011

Kickstarted 17:16 15-Jul-2011

[username@mmilcluster4]$ qlogin -l "h_vmem=4g"
Your job 172119 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 172119 has been successfully scheduled.

Establishing /opt/gridengine/bin/lx26-amd64/qlogin_wrapper session to host mmil-  compute-4-
28.local ...
Warning: Permanently added 'mmil-compute-4-28.local' (RSA) to the list of known hosts.

Rocks Compute Node
Rocks 5.4 (Maverick)
Profile built 11:43 12-Mar-2012

Kickstarted 11:49 12-Mar-2012

[username@mmil-compute-4-28 ~]$ mat

                            < M A T L A B (R) >
                  Copyright 1984-2010 The MathWorks, Inc.
                Version 7.12.0.635 (R2011a) 64-bit (glnxa64)
                               March 18, 2011

  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, visit www.mathworks.com.

>> exit
```

or...

```
[username@mmil-compute-4-28 ~]$ matlab
```



Checking Status

Before submitting any jobs to a cluster, first check the cluster's status. There may already be a vast number of jobs ahead of you in the queue. In this case, it may be a while before any newly submitted jobs actually start running, depending on the type of data processing being done (refer to Average Job Run Times). If many jobs are running on one cluster, you may log in to another cluster running fewer jobs.

To check the status of a cluster's job queue, log in to that cluster and type one of the following commands:

$ qstat (lists all cluster jobs submitted by the current user)
$ qstat –u "*" (lists all cluster jobs submitted by any user)
$ qstat – r (lists details of cluster jobs)
$ qstat | wc (counts the number of cluster jobs in the queue)
$ qmon (starts GUI that displays pending, running, and finished cluster jobs)

You may also periodically check the status of submitted jobs to determine their progress and to confirm their completion. When a processing job encounters an error, the job will quit before finishing the entire process and the cluster job queue does not provide an obvious sign that the job terminated early. For this reason, it is important to check the output and error files (details in Troubleshooting) to verify successful completion of all jobs.

Creating Jobs

Jobs are script files that are created as an output of certain c-shell and matlab MMPS scripts. For example, a commonly used matlab MMPS script is:

```
> MMIL_Process_Exams('PROJ_ID')
```

After this MATLAB process has finished (more info on running MATLAB), a command is printed onscreen:

```
qmatjobs PROJ_ID_MMIL_Process_Exams
```

If a c-shell MMPS script was run to create the jobs, you will instead see the following command printed onscreen:

```
qcshjobs PROJ_ID_MMIL_Process_Exams
```

Either line indicates that the jobs have been created. The job files, in this example, reside in the directory:

```
/home/username/batchdirs/Proj_ID_MMIL_Process_Exams
```

The directory name is determined by the project and MMPS script that was used to create the job files. Also within this directory (more details in Directory Structure) is a *scriptlist.txt* file, which contains a list of all jobs to be processed. When submitting jobs, this file is read first and all jobs are included by default. To control which jobs will be run, you may edit the *scriptlist.txt* file to list only the desired jobs.

Submitting Jobs

Once the jobs have been created according the previous section, copy the command printed in the terminal (e.g. qmatjobs ***Proj_ID***_*MMIL_Process_Exams*) and paste into a terminal that is logged in to one of the clusters. The terminal will print onscreen that the jobs have been submitted. It is recommended that you then check the status of your jobs to confirm that they've been added to the job queue.

You may be instructed to use a variation of the *qmatjobs* or *qcshjobs* command, such as *qmatjobs3* or *qmatjobs6,* indicating that the processes require additional memory (RAM) to complete. By default, 2 GB of RAM are allocated per job. The command *qmatjobs2* will allocate 4 GB, *qmatjobs3* will allocate 6 GB, and so on; the maximum amount of RAM that can be allocated for a single job node is 16 GB.

Average Job Run Times

The time required for a single job to complete primarily depends on the type of data being processed and the stage of data processing. Below are estimates for a single MRI session (**include MEG?**) at various steps of the processing pipeline:

- Process_Exams
  - Structural Only: 1 hour
  - Structural + DTI: 1-2 hours
  - with DTI_ATLflag on: 2-4 hours
- FSRecon_Exams:
  - First run: 24 hours
  - Rerunning after edits: 12 hours
  - IcoResamp: 30 minutes
- Analyze_Exams:
  - MRI: 1 hour
  - DTI: 1.5 hours
  - Longitudinal: 1 hour
- Summarize (both MRI and Long): 10 seconds
- Long_Setup_Exams: 10 minutes
- Long_Register_Exams: 2 hours

Troubleshooting

This section contains details to assist you in diagnosing errors that you may encounter while using the MMIL clusters to process data.

Output and Error Files

The output (*.out) and error (*.err) files are located in the */pbsout* sub-directory within the directory containing the respective job files (see Creating Jobs). The **.out* file is a record of output from the entire process and can be quite long. The **.err* file is a record of errors that have interrupted the processing. (There are exceptions; running *Long_Setup_Exams* will print all output to the **.err* file.) Warnings and errors - when they occur – may be found in both files, so it's recommended that you inspect them even if the data processing completed successfully. Continuing with the example, the output files created by running *qmatjobs* ***PROJ_ID*** *_MMIL_Process_Exams* on the cluster are located here:

```
/home/username/batchdirs/Proj_ID_MMIL_Process_Exams/pbsout
```

To view the files independently, navigate to this directory and type at the command line:

```
$ cat job_001_SubjID.out
$ cat job_001_SubjID.err
```

You may also use a text editor to view these files:

```
$ nedit job_001_SubjID.out
```

The *.*out* files can be quite long, and error messages usually occur at the end of the file. Consequently, you may read only the last few lines of the file by typing:

```
$ tail job_001_SubjID.out
```

Alternatively, you may search all files for a certain term. Key terms in these files that may indicate a processing problem are *ERROR* and *WARNING*. For instance, to return a list of all output files that contain the term *ERROR*:

```
$ grep *.out ERROR
$ grep *.err ERROR
```

When errors are found,

Resubmitting Jobs

If a job quits soon after it is submitted, an error may have been encountered. Check the output and error files (details in previous section).

Local Processing

To submit jobs for processing by the local computer's processor, the same rules apply as Submitting Jobs on the MMIL cluster, however the commands used are *xmatjobs* and *bmatjobs*.