

# HAMMER

## User's Manual

Section of Biomedical Image Analysis

© *SBIA, University of Pennsylvania, April 15, 2008*



# Contents

<b>1</b>	<b>HAMMER Overview</b>	<b>7</b>
1.1	Description . . . . .	7
1.1.1	Convention for segmented images . . . . .	7
1.1.2	Notes on preprocessing images for HAMMER . . . . .	8
1.2	Software Dependencies . . . . .	9
1.3	Licensing . . . . .	9
1.4	Features . . . . .	10
<b>2</b>	<b>Obtaining HAMMER</b>	<b>13</b>
2.1	Downloads . . . . .	13
2.2	Installation . . . . .	13
2.2.1	Installation on a Windows system . . . . .	13
2.2.2	Installation on a Linux system . . . . .	14
<b>3</b>	<b>Getting Started with HAMMER</b>	<b>17</b>
3.1	Using HAMMER CLI on the sample data-set . . . . .	17
3.1.1	Labeling the sample subject's brain using the included template/models . . . . .	18
3.1.2	Normalizing the sample subject's brain using the included templates/models . . . . .	19
3.1.3	Generating the RAVENS maps for the sample subject's brain	19
3.1.4	Correct the labels generated in the labeling step . . . . .	19
3.2	Using HAMMER GUI on the sample data-set . . . . .	20

3.2.1	Description of the sample data-set . . . . .	20
3.2.2	Creating a project for the sample data-set . . . . .	20
3.2.3	Using the project properties panel . . . . .	24
3.2.4	Labeling the images in the data-set . . . . .	24
3.2.5	Correcting labels . . . . .	24
3.2.6	Normalizing the images in the data-set . . . . .	24
3.2.7	Generating RAVENS maps . . . . .	25
3.2.8	Anatomy of the HAMMER GUI . . . . .	25
<b>4</b>	<b>Using the HAMMER CLI</b>	<b>31</b>
4.1	Labeling a subject's brain by a manually-labeled brain model . . . . .	32
4.1.1	Required arguments . . . . .	32
4.1.2	Valid options . . . . .	32
4.1.3	Examples . . . . .	33
4.1.4	Notes . . . . .	33
4.2	Normalizing subject brain images . . . . .	34
4.2.1	Required arguments . . . . .	34
4.2.2	Valid options . . . . .	34
4.2.3	Examples . . . . .	35
4.2.4	Notes . . . . .	35
4.3	Generating RAVENS maps for each tissue (WM, GM, VN) . . . . .	36
4.3.1	Required arguments . . . . .	36
4.3.2	Valid options . . . . .	36
4.3.3	Examples . . . . .	37
4.3.4	Notes . . . . .	37
4.4	Group Analysis . . . . .	38
<b>5</b>	<b>Using the HAMMER GUI</b>	<b>39</b>
5.1	Organizing Data and Starting HAMMER with your Own Project . . . . .	39
5.2	Using Projects to Group Subjects . . . . .	39

<i>CONTENTS</i>	5
5.3 Browsing the files associated to subjects in a project . . . . .	41
5.4 Perform Any Processing Step . . . . .	42
5.5 Help . . . . .	43
<b>References</b>	<b>45</b>



# 1

## HAMMER Overview

### 1.1 Description

HAMMER is an elastic registration procedure, with details in [1], [2], [3]. It is used to warp a template brain image to an individual, and vice versa. HAMMER is provided in two modalities: A HAMMER Command Line Interface (CLI) and a HAMMER Graphic User Interface (GUI). Both offer a way to perform common tasks using the HAMMER algorithm. The Command Line Interface allows users to submit jobs via job schedulers, write their own scripts to perform batch operations, etc., besides being able to simply run HAMMER on the command line. On the other hand the Graphic User Interface allows users to perform repetitive operations for a set of subjects with the same options with a few clicks of a mouse.

Throughout this document it is assumed that you intend to perform a warping task on images that have gone through a minimal series of preprocessing steps described in the next section. Ultimately, the images must adhere to the following convention.

#### 1.1.1 Convention for segmented images

In order to make HAMMER as robust as possible to different acquisition protocols and conditions, we provide a distribution that assumes that images have been skull-stripped and segmented into gray matter (GM), white matter (WM), and ventricular CSF. A variety of other software tools that can perform these steps, including skull stripping, reorientation and reslicing, and segmentation tools exist, some of which can be provided by our laboratory upon request. We use an image intensity of 250 for WM, 150 for gray matter GM, 50 for ventricles and 10 for CSF in the tissue-

segmented brain images. Alternatively if the model uses the same value for the entire CSF, including the ventricles, the same consideration should be allowed with each individual image. However, in our provided model, ventricular CSF is labeled as 50 and it does make the registration more accurate; for consistency, ventricular CSF in the subjects should be also identically labeled as 50.

### 1.1.2 Notes on preprocessing images for HAMMER

Raw MR images should be subject to some preprocessing steps before running the HAMMER algorithm on these. The steps may include reorientation, alignment, skull stripping and tissue segmentation. Once these images have been properly pre-processed then it is possible to use HAMMER to obtain the labeling results or to produce tissue density maps such as RAVENS maps [ref for voxel-based analysis] or to normalize subject images to a template/model image. RAVENS maps are generated by using both the HAMMER-output deformation field and the tissue-segmented brain image as inputs. The results of performing the HAMMER algorithm can then be used to perform a group analysis by using a statistical analysis method such as SPM.

#### Skull Stripping

First of all, please check the orientation of the image. The required input image should be in Analyze-format following the radiology convention (left hemisphere on the right side). The header file must be included in addition to the image file. Programs like MIPAV can be used to reorient the images.

As for skull-stripping of original MR brain images, any of the following free-software can be used, if it produces a reasonable result for your data. These include: BET, which is a part of FSL , Brain Suite (from USC), MIPAV with free skull-stripping plug-in.

The cerebellum can also be removed. We have found that removal of the cerebellum yields better results in the medial temporal lobe. If the user prefers to retain it he/she must use a template with cerebellum.

#### Tissue segmentation

There are a lot of programs available for tissue segmentation, such as FAST (as part of FSL) and FANTASM. Different segmentation programs use different labels for WM, GM, and CSF. It is recommended to follow our aforementioned segmented



image conventions (250 for WM, 150 for GM, and 10 for CSF), however you may use your own if both, your subject and model follow the same conventions.

The ventricular CSF is labeled with a value of 10, after previous segmentation. The next step typically segments out ventricular CSF and then labels it by using the value of 50.

NOTE: In order to label the ventricle correctly, it is recommended that both the input image and the model have dimensions 256x256 in XY plane and it must have the same orientation. MIPAV can be used to crop or pad the images.

## 1.2 Software Dependencies

This piece of software works on a variety of Linux and Windows systems with JRE 1.5 or higher being the only extra requirement. However, for linux systems it is recommended to have FSL 3.3.2 or higher installed.

Additionally, users may find extremely useful to have visualization tools to view the results.

Note that even though a version of FSL exists for Windows systems, this does not completely integrate with the Windows operative systems, instead it runs on top of cygwin, a Linux emulator layer. As a consequence it has been decided not to integrate HAMMER with FSL on Windows platforms.

## 1.3 Licensing

By using software developed by The Section of Biomedical Image Analysis (SBIA) at The University of Pennsylvania, you are agreeing to the following terms and conditions:

1. Permission is granted to use this software without charge for non-commercial research purposes only.
2. SBIA retains all rights and ownership of any software developed by this laboratory.
3. You may not redistribute the obtained software. You may create copies for your own personal use but we ask you to refer any one else to SBIA's website (<https://www.rad.upenn.edu/sbia/software/index.html>) for obtaining a copy of our software.

4. You may not remove or alter any copyright, license or other proprietary notices in the software.
5. You may publish papers or books using results produced using SBIA's software always providing appropriate citations.
6. The software is provided "AS IS". None of the authors or members of The Section of Biomedical Image Analysis at the University of Pennsylvania assume any liability or responsibility for the use of this software.
7. In particular, the software is for research purposes only and it has not been evaluated for clinical use.

## 1.4 Features

Currently the HAMMER CLI and the HAMMER GUI can be used to perform the following tasks:

- Label a subject's brain using a model
- Normalize a subject's brain to a given model
- Computation of RAVENS maps of a subject's brain
- Correct labels

Additionally, the HAMMER GUI allows you to group multiple subjects into a *project* where options that apply to all the subjects in a project are specified only once, and then the operations mentioned above are performed on all the subjects sequentially.

Both, the CLI and the GUI allow the user to specify, the path where the temporary files are to be written. This could be important for example when access to local disks is faster as supposed to networked disks. This feature also helps to keep intermediate results some what controlled, as these can occupy large disk space overtime.

The CLI has a *clean* function that simply deletes all the forgotten intermediate results in the default path: *user's home/.hammer/*. The GUI however always cleans up after itself before exiting, unless instructed otherwise via the *keep intermediate results* option.

The intermediate results include log files that can be useful in tracking erroneous processes; the CLI produces one log file per action, while the GUI produces

one log file per session. It is important to note again that this log file will be deleted together with all the other intermediate results on clean up, unless the option to keep intermediate results is used.



# 2

## Obtaining HAMMER

### 2.1 Downloads

- Linux and Windows distributions: <https://www.rad.upenn.edu/sbia/software/index.html#hammer>
- Sample data set (For the CLI): <https://www.rad.upenn.edu/sbia/distributions/hammer/sample/sample.zip>
- Sample data set (For the GUI): [https://www.rad.upenn.edu/sbia/distributions/hammer/sample/sample\\_batch.zip](https://www.rad.upenn.edu/sbia/distributions/hammer/sample/sample_batch.zip)
- Sample models: <https://www.rad.upenn.edu/sbia/distributions/hammer/models/models.zip>

### 2.2 Installation

HAMMER can be installed on Windows XP, Windows 2000 and Linux.

#### 2.2.1 Installation on a Windows system

Before installing HAMMER please make sure that you have previously installed Java JRE 1.5 or higher. Both the CLI and the GUI depend on it to work.

After a proper version of Java is available in the system, install HAMMER on a Windows System by simply downloading the installer [2.1], and double clicking on it. Select a path where to install the software and click OK.

## 2.2.2 Installation on a Linux system

The installation on a Linux system is a little more manual and it involves the following steps:

- Install the prerequisites:
  - If needed, install Java (JRE 1.5 or higher) following these instructions:  
<http://java.com/en/download/index.jsp> .
  - If needed, install FSL following these instructions:  
<http://www.fmrib.ox.ac.uk/fsl/fsl/downloading.html>
- Download the linux version following the link provided earlier in this document [2.1].
- Unpack the tarball in the desired application folder. Note that you may need administrator privileges in the machine if the path is not under the user's home directory.
- Set the PATH environment. This would depend on the particular shell that you are using. For example, the following lines can be added at the end of your .profile or .bashrc file, where it is assumed that you want to install everything in the folder /home/username/Applications, and 'username' is replaced by your username.

```
# The next lines add fsl to the path

FSLDIR=/home/username/Applications/fsl
. ${FSLDIR}/etc/fslconf/fsl.sh
PATH=${FSLDIR}/bin:${PATH}
export FSLDIR PATH

# The next lines add HAMMER to the path

HAMMERDIR=/home/username/Applications/hammer
PATH=${HAMMERDIR}/hammergui/bin:${PATH}
PATH=${HAMMERDIR}/bin:${PATH}
export HAMMERDIR PATH
```

It is strongly recommended that you go over the section [3] that describes how to get started using the sample data referenced in the downloads section [2.1].

This will not only check that your system is ready to run HAMMER, but also will allow you to become familiar with the basic assumptions for images and project structures.





# 3

## Getting Started with HAMMER

This section shows you in detail how to get started with HAMMER using the sample data that can be downloaded from the web [2.1], the detailed steps here make use of the default options, if you wish to obtain more information about each of the options please refer to the *Using HAMMER* section [4].

Depending on your interests you may wish to review either or both of the following sections that describe how to get started using the CLI and the GUI.

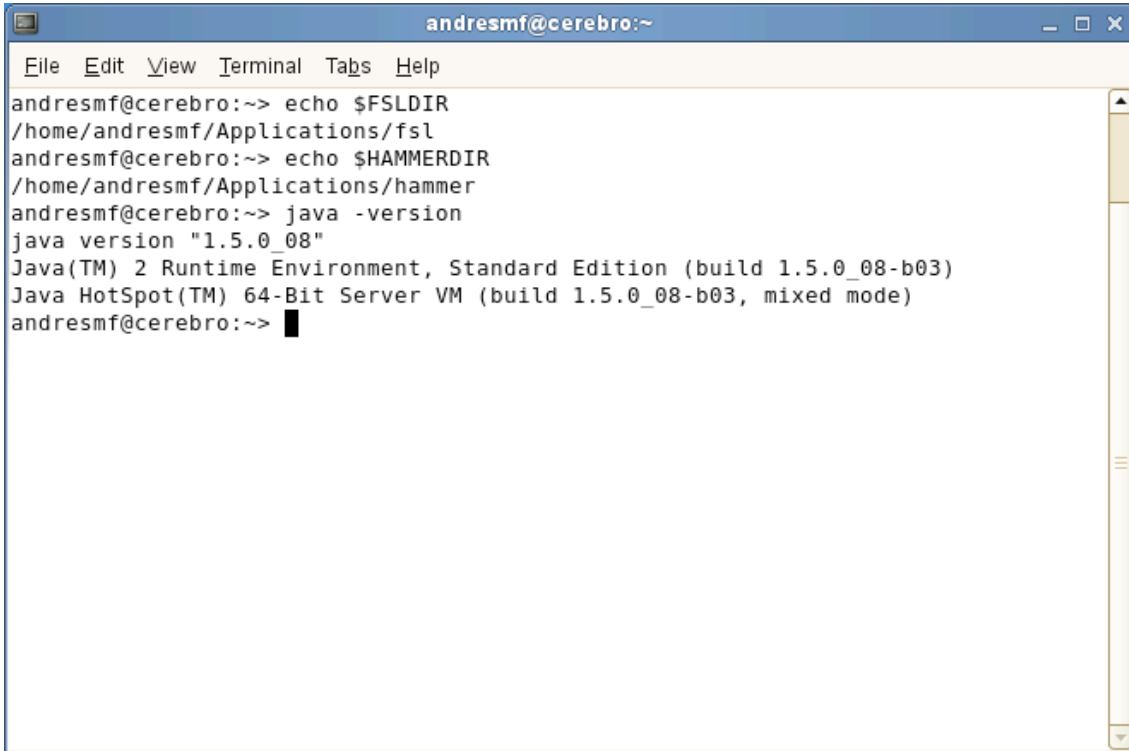
For the CLI use the data-set that includes only one subject, for the GUI it is recommended that you use the data-set that includes a few subjects so that you understand how the GUI expects you to organize your subjects and to get familiar with the project processing.

### 3.1 Using HAMMER CLI on the sample data-set

Before running HAMMER with your subjects take a moment to run HAMMER using the sample data provided [2.1]. Also, make sure that you have a working installations of Java 1.5 or higher by typing at the shell: `java -version` . Update your version of java if needed or if you have already installed Java 1.5 or later then make sure that you edit your Java PATH in `.profile` or `.bashrc` file. FSL is recommended and you can test if is properly installed by typing in the shell: `fsl` . Figure [3.1] shows a terminal where the user checks for the environment configuration

It is recommended that you run a test for each of the following four tasks:

- Label the sample subject's brain using the included template/models
- Normalize the sample subject's brain using the included templates/models

A terminal window titled 'andresmf@cerebro:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal output shows the following commands and results:

```
andresmf@cerebro:~> echo $FSLDIR
/home/andresmf/Applications/fsl
andresmf@cerebro:~> echo $HAMMERDIR
/home/andresmf/Applications/hammer
andresmf@cerebro:~> java -version
java version "1.5.0_08"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_08-b03)
Java HotSpot(TM) 64-Bit Server VM (build 1.5.0_08-b03, mixed mode)
andresmf@cerebro:~> █
```

Figure 3.1: A terminal screen showing that my environment is properly configured.

- Generate the RAVENS maps for the sample subject’s brain
- Correct the labels generated in the labeling step

In order to perform this test, download the models and sample dataset from the sources listed earlier in this document [2.1].

Unzip the contents to folder of your choice and then run the following commands to perform each of the tests accordingly.

### 3.1.1 Labeling the sample subject’s brain using the included template/models

In order to label the sample subject’s brain using the included templates you simply need to run the following command on a terminal from the directory where the sample images were unzipped. Note that if you are using a windows system you have to use the “HAMMER CLI” that should be found on your programs menu after a successful installation.

```
$ hammer labelbrain -S subject_segmented_no_cere.img  
-M jakob_rad_convention_segmented_no_cere.img  
-L jakob_rad_convention_labels_no_cere.img
```

### 3.1.2 Normalizing the sample subject's brain using the included templates/models

In order to normalize the sample subject's brain using the included template you simply need to run the following command on a terminal from the directory where the sample images were unzipped. Note that if you are using a windows system you have to use the "HAMMER CLI" that should be found on your programs menu after a successful installation.

```
$ hammer normalizebrain -S subject_segmented_no_cere.img  
-O subject_original_with_cere.img  
-M jakob_rad_convention_segmented_no_cere.img  
-L jakob_rad_convention_labels_no_cere.img
```

### 3.1.3 Generating the RAVENS maps for the sample subject's brain

To generate the RAVENS maps for the sample subject's brain using the included templates you simply need to run the following command on a terminal from the directory where the sample images were unzipped. Note that if you are using a windows system you have to use the "HAMMER CLI" that should be found on your programs menu after a successful installation.

```
$ hammer ravens -S subject_segmented_no_cere.img  
-M jakob_rad_convention_segmented_no_cere.img
```

### 3.1.4 Correct the labels generated in the labeling step

Finally to correct the label image that was generated in the first step run the following command on a terminal from the directory where the sample images were unzipped. Note that if you are using a windows system you have to use the "HAMMER CLI" that should be found on your programs menu after a successful installation.

```
$ hammer correctlabels -S subject_segmented_no_cere.img  
-L subject_segmented_no_cere.img.Warped.Labels.img  
-O subject_segmented_no_cere.img.Warped.LabelsCorrected.img
```

## 3.2 Using HAMMER GUI on the sample data-set

Before you start using the HAMMER GUI you should make sure you have it properly installed in your system. Additionally, for this section you will need the `sample_batch.zip` and `models.zip` archives containing the sample project data-set and models referenced in the downloads section [2.1].

If you installed HAMMER successfully, you should be able to start the HAMMER GUI by typing at the prompt:

```
$ hammergeui
```

Or by clicking on the *HAMMER GUI* menu entry on a Windows system.

### 3.2.1 Description of the sample data-set

The sample data set is contained in the `sample_batch.zip` (see downloads for a link to obtain this). Figure [3.2] shows the file structure of the sample project after uncompressing the archive.

The *cbq* images have gone through a minimum of preprocessing steps, such as conversion to Analyze format, ACPC alignment, padding, etc. Additionally, the *preprocessed* images have already been segmented to meet the image requirements for HAMMER [1.1.1].

Use your favorite Analyze format image viewer to take a look at the images.

### 3.2.2 Creating a project for the sample data-set

With HAMMER open, as shown in figure [3.3], follow these steps:

- If it is not already open, open a *Project Manager Window* (Window → Project Manager)
- Right click on the *Hammer Projects* folder of the *Project Manager*.

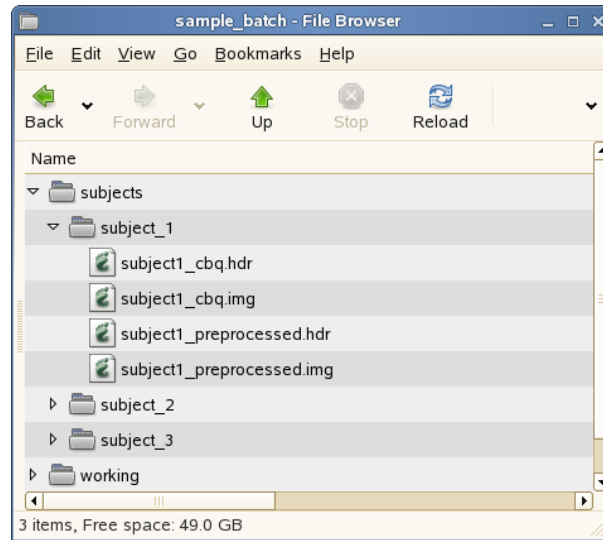


Figure 3.2: A file browser showing the structure of the batch sample project.

- Select and click on the *Add Project* option of the pop up menu.
- A *New Project Form* will appear in the main panel, this, as well as any other panel can be made to use the whole window by simply double clicking on its corresponding tab as shown on figure [3.4] (this one also displays an  $\times$  that is used to close the panel.) By double clicking on the tab a second time, the panel returns to its initial position using only a portion of the main window. As you become more familiar, you will see that the panels can be rearranged to suit your preferences.
- Fill the *New Project Form* with the the default parameters, as follows:
  - Give a name to your project. This has to be unique among the projects on the same folder and must not contain any special character that can not be included on a file name.
  - Provide the path for the `subjects` directory inside `sample_batch`, that is, the sample data set that was described earlier.
  - Provide the path for the model folder resulting from unzipping the mentioned archive. This path should contain all the models that will be used by the registration algorithms.
  - Select a working path, for example the `working` folder in figure [3.2] . This path will be used to place some temporary results and files. Log execution files will also be placed in this path. Note, a subfolder will be

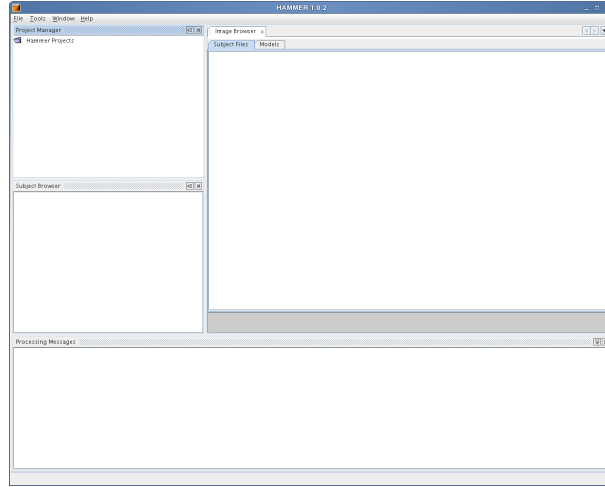


Figure 3.3: *HAMMER GUI when it is opened for the first time.*

created for each subject, and this folder will contain all the intermediate results corresponding to the processing of this subject.

- Specify the path for the segmented model that will be used during the multiple advanced processing steps. Select `jakob_rad_convention_segmented_no_cere.img` from the HAMMER distribution models.
- Specify the path for the label model that will be used during the multiple advanced preprocessing steps. Select `jakob_rad_convention_labels_no_cere.img` from the HAMMER distribution models.
- Leave the *Linear Transformation Field* empty. This field would only be used by experienced users that have already precomputed a linear transformation and they wish to use it here.
- Leave the default input suffix. This is the suffix that input images are expected to have. Other images will not be processed.
- Leave the default preprocessed suffix. This is the suffix that preprocessed images will have. Ultimately images with this suffixes will be processed by hammer. NOTE: Even if your images have been segmented already they need to be preprocessed, therefore only preprocessed images that have corresponding input images will be processed by hammer.
- Leave the default neighborhood search sizes. The neighborhood search size is different at each resolution. The defaults are: `<12, 10, 8>` mm/voxels for `<value_at_low, value_at_middle, value_at_high>`
- Set to `<10, 10, 10>` the number of warping iterations for low, mid and high resolutions. This is the number of iterations in each of the warping

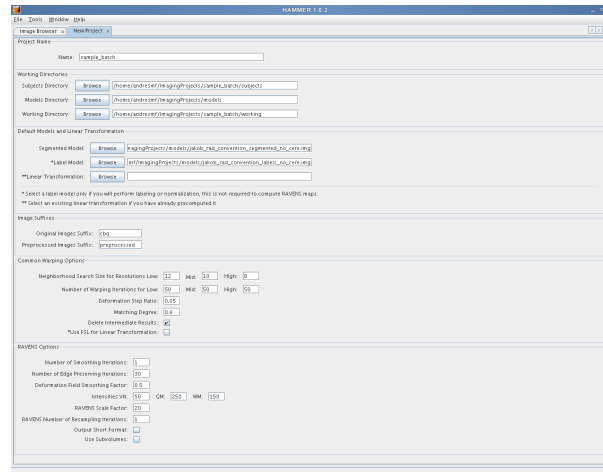


Figure 3.4: *The New Project Form.*

stages. It is recommended that your use  $<50, 50, 50>$  for best results but for the purpose of illustrating the functionality, 10 iterations in each stage are good enough and would take less time to process.

- Leave the default deformation step ratio. This is the step ratio used in the warping step.
- Leave the default matching degree. This is the similarity degree that will be used during the warping step.
- Leave the default keep intermediate results option.
- If you are using Linux and have FSL installed you can select the FSL option.
- Leave the default number of smoothing iterations. This is the number of smoothing iterations that will be used in the RAVENS maps generation.
- Leave the default number of edge preserving iterations.
- Leave the default deformation field smoothing factor.
- Leave the intensity values for VN, GM, WM. These are the values that will be assigned to ventricles, gray matter and white matter respectively.
- Leave the default ravens scale factor.
- Leave the default number of RAVENS resampling iterations.
- Click on *Create Project* to finish the project creation.

Note that the set of RAVENS options only apply if what you want to do is to compute RAVENS maps, otherwise you do not need to pay attention to them.

### 3.2.3 Using the project properties panel

The properties panel can be accessed by either right clicking on a project name (on the project tree) or by clicking on the (Window → Properties), which will display the properties of the selected project. The properties panel will allow the user to edit any of the properties of the selected project.

### 3.2.4 Labeling the images in the data-set

To perform the labeling step for the sample data set follow these steps:

- Select the sample project from the *Project Manager Window*.
- Click on (Tools → Perform Labeling Step). This will use HAMMER to warp the model to each of the subject images using the label model.

After the warping step with input `subject*_preprocessed.img` one gets the additional files shown in figure [3.5]. The subjects are warped one by one.

### 3.2.5 Correcting labels

Some times the label image created in the previous step need s some correction, to do so, simply click on (Tools → Perform Label Correction Step).

After the correction step we have the files listed on figure [3.6].

### 3.2.6 Normalizing the images in the data-set

To normalize all the subjects in the sample data-set to the segmented model follow these steps:

- Select the sample project from the *Project Manager Window*.
- Click on (Tools → Perform Normalization Step). This will use HAMMER to warp all the preprocessed subject images to the segmented model.

After running the normalization step with input `subject*_preprocessed.img` one gets the additional files shown in figure [3.7]. The subjects are warped one by one.



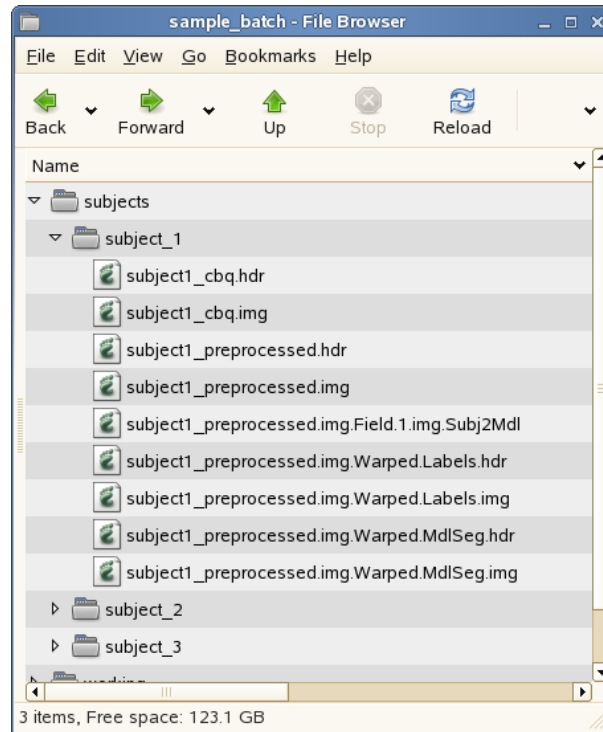


Figure 3.5: A file browser showing the structure of the batch sample project after labeling.

### 3.2.7 Generating RAVENS maps

To generate RAVENS maps follow these steps:

- Select the sample project from the *Project Manager Window*.
- Click on (Tools → Perform Ravens Map Step). This will use HAMMER to warp all the preprocessed subject images to the segmented model, and generate RAVENS maps for VN, GM, and WM.

After the process is finished we should have the additional files listed on figure [3.8]. The subjects are warped one by one.

### 3.2.8 Anatomy of the HAMMER GUI

The figure [3.9] shows you a typical view of the HAMMER GUI, showing you the different parts of the GUI using the default settings. You will quickly notice however

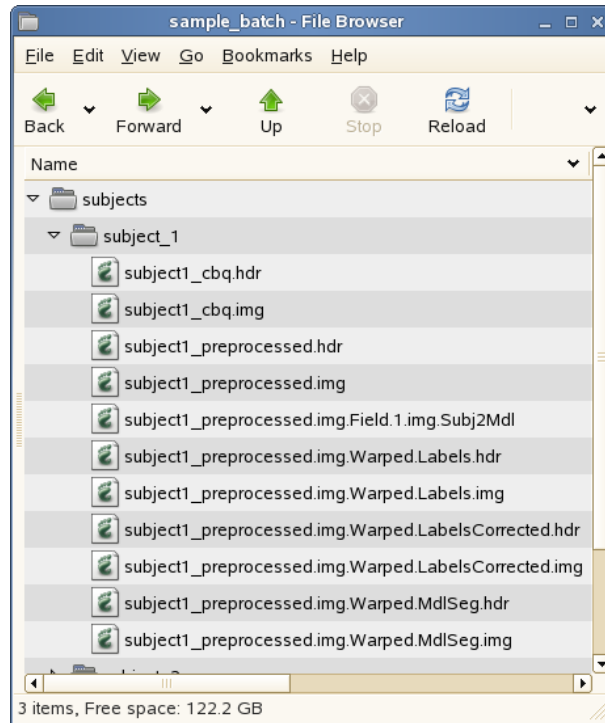


Figure 3.6: A file browser showing the structure of the batch sample project after correcting labels.

that these different parts of the GUI can be reorganized to suit your preferences. Even further, you can open and close panels by using the *Window* menu or by clicking on the cross for each of the panels.

As you will see, the HAMMER GUI works using *projects* as means of organization, for each of these projects, the user needs to specify a *subject* path, this should correspond to a directory containing a set of subdirectories, one for each of the subjects in the project. This organization allows you to keep all the files (existing and to be generated) corresponding to each subject inside its corresponding subfolder. You will see that the sample data-set is structured that way.

As shown in figure [3.9] the HAMMER GUI is provides multiple panels:

- The *Project Manager* is a panel that will display all the projects as they are created, they can be organized in folders for relevance, the name of these projects does not need to match the name of the folder where the subjects are located.
- The *Subject Browser*, simply displays the subject subdirectories of the project

currently working on. To populate this panel simply right click on an existing project and select the *Load Project* option. Clicking on a subject folder will display its contents in the *Subject Files* tab of the *Image Browser*.

- The *Image Browser* panel simply displays the contents of the selected subject's folder in the *Subject Browser*, and the models associated to the current project. Additionally, in Linux, double clicking on an image will open it with the FSL viewer if it is installed.
- The *Project Properties* panel shows the properties of the selected project. This panel is not open by default, to open it simply click on the *Properties* item of the *Window* menu, or open a floating version of it by right clicking on a project in the *Project Manager* and selecting the *Properties* item. This panel will also allow you to modify the options for the project. Note however that if you modify any of these options while HAMMER is processing, the option will only be applied to subsequent executions of it.
- Finally, the *Processing Messages* panel will display a short descriptive message of what HAMMER is currently doing. The detail messages can be found in the log files, if the user selected the keep intermediate results option.

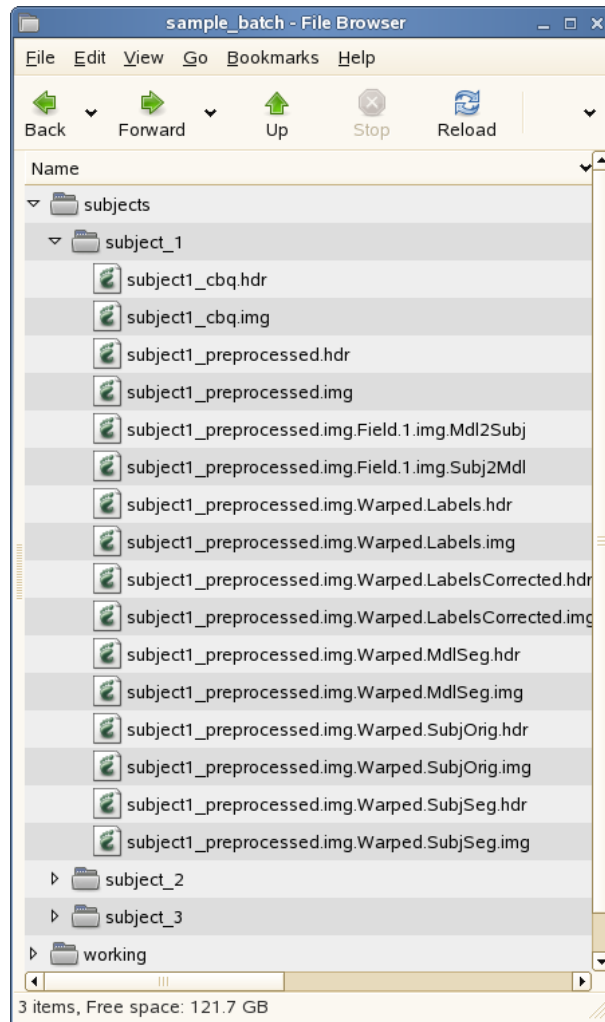


Figure 3.7: A file browser showing the structure of the batch sample project after normalization.

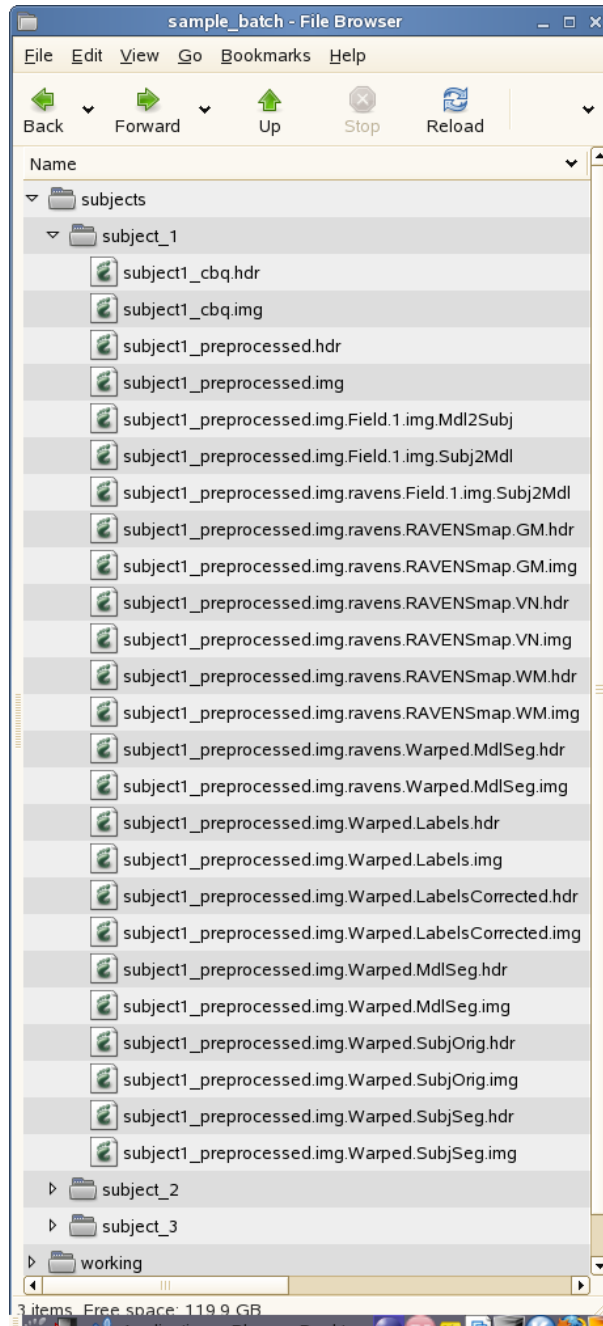


Figure 3.8: A file browser showing the structure of the batch sample project after the RAVENS maps are generated.

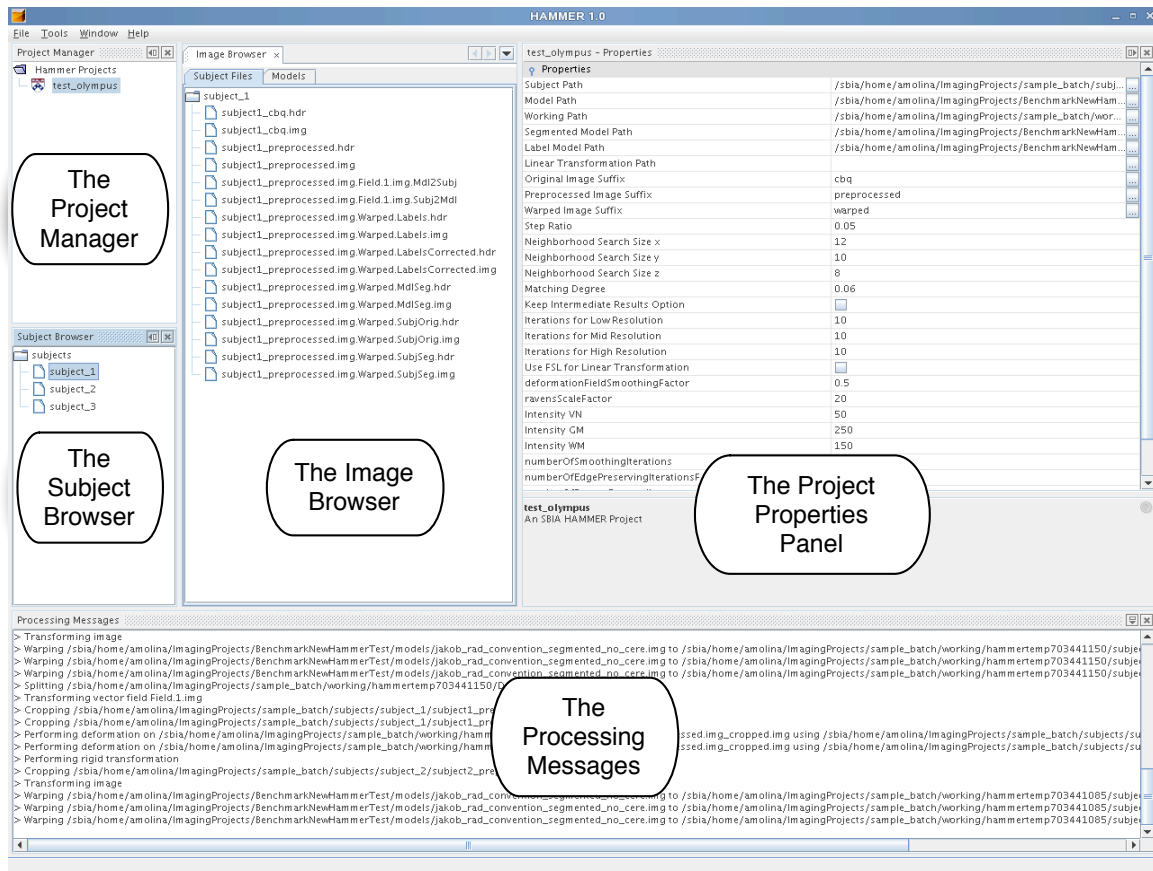


Figure 3.9: A file browser showing the structure of the batch sample project.

## 4

# Using the HAMMER CLI

This section describes in detail how to perform each of the three main tasks implemented in the HAMMER CLI: labeling a subject's brain by a manually-labeled brain model, normalizing and computation of RAVENS maps for subject brain images that have already been preprocessed to meet the brain image conventions described earlier [1.1.1].

The general syntax for the HAMMER CLI is the following:

```
hammer <subcommand> [subcommand arguments and options]
```

Where the subcommand is one of the following:

- labelbrain
- normalizebrain
- ravens

The subcommand-arguments are a set of required-arguments and options. These vary depending on the subcommand, and they will be explained next. Additionally, it is possible to run

```
hammer help <subcommand>
```

This would display help for the usage of the given subcommand.

## 4.1 Labeling a subject's brain by a manually-labeled brain model

In order to label a subject's brain by a manually-labeled brain model you simply need to run the following command:

```
hammer labelbrain -S <subject_image> -M <model_image>
-L <label_image> [options]
```

### 4.1.1 Required arguments

- **-S [string]** : Segmented subject image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- **-M [string]** : Segmented model image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- **-L [string]** : Label model image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.

### 4.1.2 Valid options

- **-A [string]** : Global affine transformation to use in the rigid transformation stage. If this is provided, then it will not be regenerated. This transformation must be in the SBIA compatible format.
- **-i [int]** : Number of iterations in the deformation field generation stages. The default is 50.
- **-R [float]** : Deformation rate. The default is 0.05



#### 4.1. LABELING A SUBJECT'S BRAIN BY A MANUALLY-LABELED BRAIN MODEL33

- `-r [int,int,int]` : Neighborhood search sizes to use in the deformation field generation stages at each resolution. Input as [value-at-low,value-at-middle,value-at-high]. The default is [12,10,8]. Each number indicates mm/voxels at the given resolution.
- `-m [float]` : Matching degree. The default is 0.06
- `-d [string]` : Full path for temporary results. This is optional but recommended in some systems. For example, it may speed the process to read and write temporary results on a local drive rather than across the network. Use in combination with `-T` option to keep the intermediate results.
- `-T`: Keep intermediate results.
- `-fsl`: Use flirt for computing the linear registration. This option is not valid in Windows systems.

#### 4.1.3 Examples

- Using the default values:

```
hammer labelbrain -S subject.img -M model.img  
-L label.img
```

- Providing some options:

```
hammer labelbrain -S subject.img -M model.img  
-L label.img -i 10 -R 0.06 -d /temp
```

#### 4.1.4 Notes

You can use your own label template, or you can use the template that comes with the package. This label model was developed by Dr. Kabani at the Montreal Neurological Institute, including 101 regions of interest. Note that, if you use Dr. Kabani's template you must acknowledge it using the following reference [ref]. This document provided a link earlier to get the model. Use the images with or without cerebellum to match your particular situation.

## 4.2 Normalizing subject brain images

In order to normalize the subject image to the space of the model image run the following command:

```
hammer normalizebrain -S <segmented_subject_image>  
-O <original_subject_image> -M <segmented_model_image>  
-L <label_image> [options]
```

### 4.2.1 Required arguments

- **-S [string]** : Segmented subject image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- **-O [string]** : Original subject image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- **-M [string]** : Segmented model image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- **-L [string]** : Label model image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.

### 4.2.2 Valid options

- **-A [string]** : Global affine transformation to use in the rigid transformation stage. If this is provided, then it will not be regenerated. This transformation must be in the SBIA compatible format.

- `-i [int]` : Number of iterations in the deformation field generation stages. The default is 50.
- `-R [float]` : Deformation rate. The default is 0.05
- `-r [int,int,int]` : Neighborhood search sizes to use in the deformation field generation stages at each resolution. Input as [value-at-low,value-at-middle,value-at-high]. The default is [12,10,8]. Each number indicates mm/voxels at the given resolution.
- `-m [float]` : Matching degree. The default is 0.06
- `-d [string]` : Full path for temporary results. This is optional but recommended in some systems. For example, it may speed the process to read and write temporary results on a local drive rather than across the network. Use in combination with `-T` option to keep the intermediate results.
- `-T`: Keep intermediate results.
- `-fsl`: Use flirt for computing the linear registration. This option is not valid in Windows systems.

### 4.2.3 Examples

- Using the default values:

```
hammer normalizebrain -S segmented_subject.img  
-O original_subject.img  
-M model.img -L label.img
```

- Providing some options:

```
hammer normalizebrain -S segmented_subject.img  
-O original_subject.img  
-M model.img -L label.img -i 10 -R 0.06 -d /temp
```

### 4.2.4 Notes

Again, you can use Dr. Kabanis model with or without cerebellum, mentioned earlier. Alternatively, you can use your own model.

### 4.3 Generating RAVENS maps for each tissue (WM, GM, VN)

To register subject and model images, obtaining a deformation field and three RAVENS maps run the following command:

```
hammer ravens -S <subject_image> -M <model_image> [ options ]
```

#### 4.3.1 Required arguments

- `-S [string]` : Segmented subject image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.
- `-M [string]` : Segmented model image file name. This must be the absolute path, if it is not in the current working directory. This must be an image in Analyze format of byte type, tissue segmented and with a header. Provide the path for the image and not the header's. Note that the x and y dimensions must be equal.

#### 4.3.2 Valid options

- `-A [string]` : Global affine transformation to use in the rigid transformation stage. If this is provided, then it will not be regenerated. This transformation must be in the SBIA compatible format.
- `-i [int]` : Number of iterations in the deformation field generation stages. The default is 50.
- `-R [float]` : Deformation rate. The default is 0.05
- `-r [int,int,int]` : Neighborhood search sizes to use in the deformation field generation stages at each resolution. Input as [value-at-low,value-at-middle,value-at-high]. The default is [12,10,8]. Each number indicates mm/voxels at the given resolution.
- `-m [float]` : Matching degree. The default is 0.06
- `-c [float]` : Deformation field smoothing factor. The default is 0.5

- `-C [int]` : Number of smoothing iterations. The default is 1
- `-E [int]` : Number of edge preserving smoothing iterations. The default is 30
- `-l [int,int,int]` : Intensities for [WM,GM,VN]. The default is [150,250,50]
- `-F [int]` : RAVENS scale factor. The default is 20
- `-a [int]` : Number of RAVENS re sampling iterations. The default is 1
- `-d [string]` : Full path for temporary results. This is optional but recommended in some systems. For example, it may speed the process to read and write temporary results on a local drive rather than across the network. Use in combination with `-T` option to keep the intermediate results.
- `-T`: Keep intermediate results.
- `-fsl`: Use flirt for computing the linear registration. This option is not valid in Windows systems.

### 4.3.3 Examples

- Using the default values:

```
hammer ravens -S subject.img -M model.img
```

- Providing some options:

```
hammer ravens -S subject.img -M model.img -i 10
-R 0.06 -d /temp
```

### 4.3.4 Notes

Running this subcommand would perform two tasks. It uses HAMMER to register subject brain with the model brain, and then uses the HAMMER-output deformation field and the tissue-segmented image of the subject to generate tissue-density maps (i.e., RAVENS maps) [ref]. Higher RAVENS map implies more tissue volume at a given location in the template space. Reference [ ] describes the principle of a tissue density map in detail, and describes how it can be used for voxel-based analysis. For convenience, we combine the two previous steps into a single script program. The input to this program is the tissue-segmented image, with ventricle having been labeled as 50 . The outputs are a deformation field pointing from subject to model,

and three RAVENS maps for each tissue such as WM, GM, and VN density maps. This works using other models as well, as long as the tissue segmentation is same for both model and subject, with labels 250 for WM, 150 for GM, and 10 for CSF.

## 4.4 Group Analysis

The RAVENS maps can be used for group analysis, by using an image analysis tool such as SPM. Typically these RAVENS maps are smoothed by a Gaussian or edge-preserving filter of 5-10mm, subsequently, a statistical analysis that depends on the underlying scientific question is applied to the smoothed RAVENS maps. For example, if we want to investigate local volumetric differences in GM between a group of patients and a group of normal controls, we might want to apply a voxel-wise t-test on the smoothed RAVENS GM maps.

# 5

## Using the HAMMER GUI

### 5.1 Organizing Data and Starting HAMMER with your Own Project

HAMMER uses suffixes to determine what type of treatment gets each file. The user specifies these suffixes when creating a HAMMER project.

In particular there is an *input suffix*, which in the example was chosen as *cbq*. That means that once it is specified in the options that this is the suffix for the input images, then only images that look like `myImageName_cbq.img` and their derived will be processed while performing the HAMMER steps.

It is assumed that all the subjects to be processed with the same settings are in one folder in the user's system, preferably organized in subfolders so that only one input image is in each folder. This last thing is not a requirement but it helps the HAMMER GUI to keep things organized. In particular when browsing the files by subject, the Hammer Suite will assume that there is a one-to-one correspondence between subfolders and subjects.

### 5.2 Using Projects to Group Subjects

The HAMMER GUI is structured to group subjects and settings into projects. These projects can be further grouped into folders.

1. To create a new project follow these steps:
  - If it is not already open, open a *Project Manager Window* (Window →

Project Manager)

- Right click on the *Hammer Projects* folder of the *Project Manager*.
- Select and click on the *Add Project* option of the pop up menu.
- Fill the *New Project Form* with your project parameters, as follows:
  - Give a name to your project. This has to be unique among the projects on the same folder and must not contain any special character that can not be included on a file name.
  - Provide the path for the `subjects` directory inside `sample_batch`, that is, the sample data set that was described earlier.
  - Provide the path for the model folder resulting from unzipping the mentioned archive. This path should contain all the models that will be used by the registration algorithms.
  - Select a working path, for example the `working` folder in figure [3.2]. This path will be used to place some temporary results and files. Log execution files will also be placed in this path.
  - Specify the path for the segmented model that will be used during the multiple advanced processing steps. Select `jakob_rad_convention_segmented_no_cere.img` from the HAMMER distribution models.
  - Specify the path for the label model that will be used during the multiple advanced preprocessing steps. Select `jakob_rad_convention_labels_no_cere.img` from the HAMMER distribution models.
  - Specify the linear transformation path if you have pre-computed it before, else leave it blank. This feature would allow you to reuse a previously computed transformation.
  - Specify the input suffix. This is the suffix that original images are expected to have. Other images will not be processed.
  - Specify the preprocessed suffix. This is the suffix that preprocessed images will have. Ultimately images with this suffixes will be processed by hammer.
  - Specify the neighborhood search sizes. The neighborhood search size is different at each resolution. The defaults are: `<12,10,8>` mm/voxels for `<value_at_low,value_at_middle,value_at_high>`
  - Specify the number of warping iterations for low, mid and high resolutions. This is the number of iterations in each of the warping stages. It is recommended that your use `<50,50,50>` for best results.
  - Specify the deformation step ratio. This is the step ratio used in the warping step.
  - Specify the matching degree. This is the similarity degree that will be used during the warping step.



### 5.3. BROWSING THE FILES ASSOCIATED TO SUBJECTS IN A PROJECT41

- Select whether to delete the intermediate results or not.
- If you are using Linux and have FSL installed you can select the FSL option. NOTE: Even if you have installed FSL on Windows this option is not supported. FSL relies on cygwin to work which in particular simulates a UNIX-like file system which is not compatible with Windows.  
If you intend to compute RAVENS maps also:
  - Specify the number of smoothing iterations. This is the number of smoothing iterations that will be used in the RAVENS maps generation.
  - Specify the number of edge preserving iterations.
  - Specify the deformation field smoothing factor.
  - Specify the intensity values for VN,GM,WM. These are the values that will be assigned to ventricles, gray matter and white matter respectively.
  - Specify the ravens scale factor.
  - Specify the number of RAVENS resampling iterations.
  - Select if you want the output in short format and if you want to use subvolumes during the computation.
  - Click on *Create Project* to finish the project creation.

2. To load the files of project follow these steps:

- If it is not already open, open a *Project Manager Window* (Window → Project Manager)
- Select a project and right click on it.
- Select and click on the *Load Project Files* option of the pop up menu.

3. To delete a project follow these steps:

- If it is not already open, open a *Project Manager Window* (Window → Project Manager)
- Select a project and right click on it.
- Select and click on the *Delete* option of the pop up menu.

## 5.3 Browsing the files associated to subjects in a project

To populate the *Image Browser Window* follow these steps:

- Load project files
- If it is not already open, open a *Subject Browser Window* (Window → Subject Browser)
- Select any subject on the *Subject Browser Window*. This will populate the *Image Browser Window* with the images found in the selected subject's folder.

## 5.4 Perform Any Processing Step

To perform any of the processing steps in the pipeline follow these steps:

- Select a project from the *Project Manager Window*.
- Click on (Tools → Processing Action). This will schedule the selected processing over all the subjects in a project using the project options. Replace the *Processing Action* by any of the following:
  - *Perform Labeling Step* to warp the segmented model to each of the subjects in the set, assuming that the subject images have been properly preprocessed.
  - *Perform Normalization Step* to warp each of the subjects in the set to the segmented model, assuming that the subject images have been properly preprocessed.
  - *Perform Ravens Map Step* to generate RAVENS maps for each of the subjects in the set for VN, GM and WM , assuming that the subject images have been properly preprocessed.
  - *Perform Label Correction Step* to perform only the label correction step for each of the subjects in the set, assuming that the labeling step has already been performed.

## **5.5 Help**

For help contact: [sbia-software@uphs.upenn.edu](mailto:sbia-software@uphs.upenn.edu).



# References

- [1] D. Shen and C. Davatzikos. **HAMMER: Hierarchical attribute matching mechanism for elastic registration** IEEE Transactions on Medical Imaging, vol. 21, pp. 1421-1439, 2002.
- [2] D. G. Shen and C. Davatzikos **Very high resolution morphometry using mass-preserving deformations and HAMMER elastic registration** NeuroImage, vol. 18, pp. 28-41, 2003.
- [3] C. Davatzikos, A. Genc, D. Xu, and S. M. Resnick. **Voxel-Based Morphometry Using the RAVENS Maps: Methods and Validation Using Simulated Longitudinal Atrophy** NeuroImage, vol. 14, pp. 1361-1369, 2001.