
MASS Software Manual

Release v1.1 (r240)

Jimit Doshi

August 20, 2014

Contents

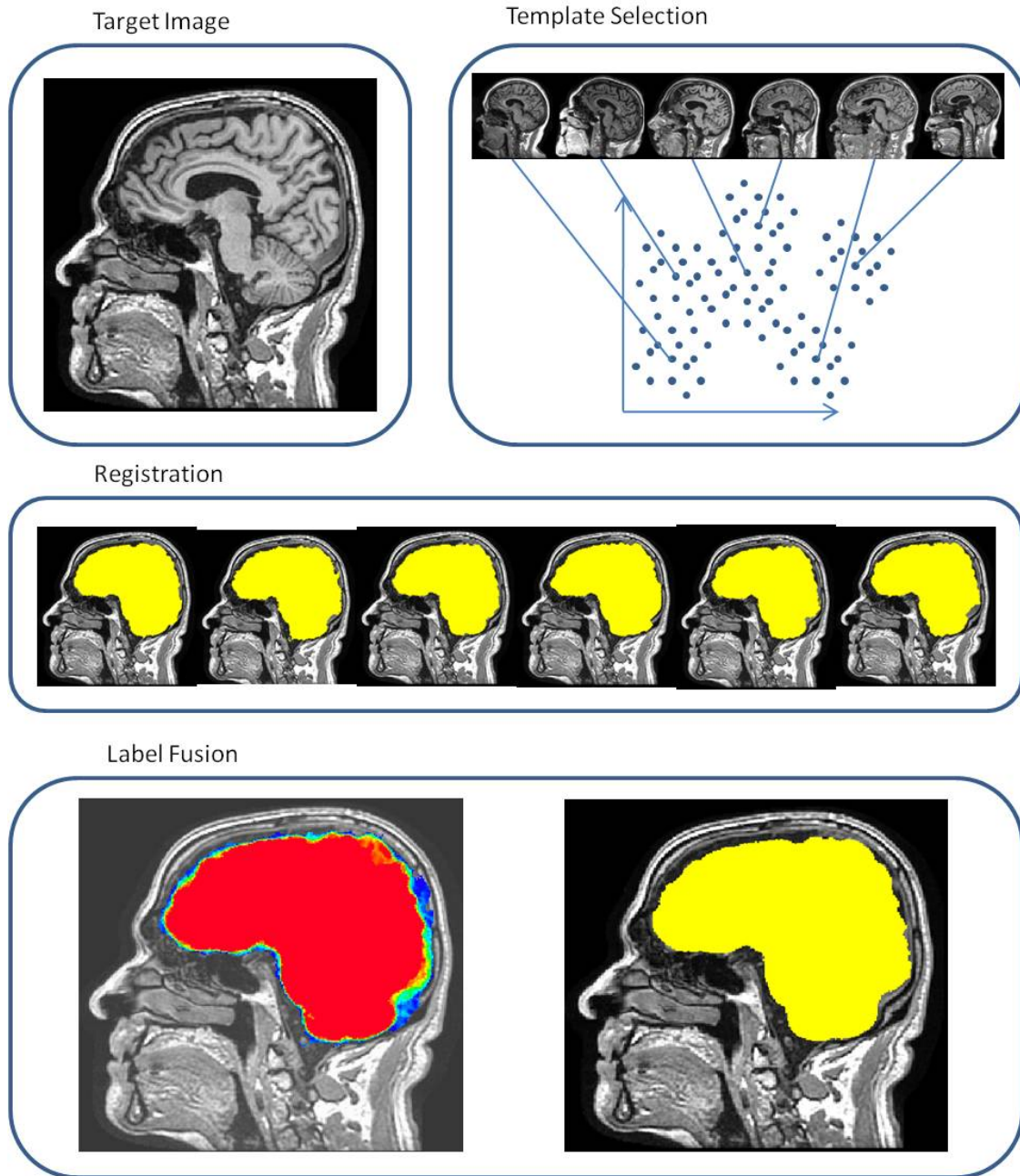
1	About the Algorithm	4
1.1	Template Selection	4
1.2	Registration	5
1.3	Label Fusion	5
2	Download	6
2.1	Software License	6
2.2	Documentation	6
2.3	System Requirements	6
2.4	Register for Download	6
3	Installation	7
3.1	Prerequisites	7
3.2	Job Scheduler	7
3.3	Configure	7
3.4	Build	8
3.5	Test	8
3.6	Install	8
4	Manual	9
4.1	General Processing Pipeline	9
4.2	Template Selection	9
4.3	MASS Default Command	10
4.4	MASS Options	10
4.5	Threshold Jacobian Rank Mask	11
5	Publications	12
5.1	Methodology	12
5.2	Validation	12
6	People	13
6.1	Advisors	13
6.2	Software Development	13
6.3	Contributors	13
6.4	Testers	13

Multi Atlas Skull Stripping (MASS) [ARAD2013], is a software package designed for robust and accurate brain extraction, applicable for both individual as well as large population studies.

MASS is implemented as a Unix command-line tool. It is fully automatic and easy to use — users input an image, and MASS will output the extracted brain and the associated brain mask.

1 About the Algorithm

A general overview of the proposed method is given in the following figure. The MASS framework consists of 3 components: template selection, registration and label fusion.



1.1 Template Selection

The quality of a registration is directly related to the similarity between the template and the target images. Either due to differences between populations (e.g. age, disease, etc.) or changes in scanner type, technology and protocol (e.g.

1.5T to 3T), images from two different projects might be significantly different. In order to increase the template-subject similarity, and hence to improve the registration accuracy, we select a study-specific set of templates using a clustering-based approach. The same set of templates is used for processing all images in the study. In this way, we limit the work required for the preparation of the ground-truth brain masks, while using templates as similar as possible to the subjects in the study.

1.2 Registration

We have chosen a recently developed publicly available registration method DRAMMS because of its ability to meet two major challenges specific to registering raw brain MR images. The first major challenge is the large amount of intensity inhomogeneity and background noise in raw brain MR images. DRAMMS finds voxel-wise correspondences by looking at multi-scale and multi-orientation Gabor texture features around each voxel. Therefore, it is relatively robust to inhomogeneity and noise. The second major challenge in registering brain MR images with skull is the possible presence of outlier regions. Outlier regions, or missing correspondences, usually refer to regions that exist in one image but not in the other. For instance, the MR image of one subject may contain more neck regions, or may have part of superior skull missing due to different field-of-view (FOV) during MRI acquisition. DRAMMS meets this challenge using the mutual salience weighting, as it adaptively finds and relies on voxels/regions that are more likely to establish reliable correspondences across images. This way, it reduces the negative impact of outlier regions compared to other registration methods that forces matching for all voxels/regions.

1.3 Label Fusion

We adopt a spatially adaptive fusion strategy that takes into consideration the local similarities between the templates and the target image. At each voxel, a weight is assigned to each template such that a higher confidence is given to templates that are locally more similar, e.g. more easily mapped, to the target image. Our main premise here is that the Jacobian maps are good indicators of local similarities between source and target images. Large Jacobian values often correlate with large geometric differences between template and target images. It's preferable to assign high weights to labels from masks that are locally similar to the subject image, as we have more confidence on the registration when the source and target images are more similar. Such a weighting mechanism is also efficient for making the method more robust. If the registration of one (or a few in the extreme case) template completely fails, the corresponding Jacobian map will have extreme values in most voxels. Thus the brain mask from this template will be ranked very low in general, and the template will not have any effect on the final extraction/segmentation.

2 Download

2.1 Software License

The MASS software is freely available under a BSD-style open source license that is compatible with the Open Source Definition by [The Open Source Initiative](#) and contains no restrictions on use of the software. The full [license](#) text is included with the distribution package and available online.

2.2 Documentation

[MASS Manual](#): Online version of this manual.

2.3 System Requirements

Operating System: Linux

2.4 Register for Download

Please [register online](#) to receive an email with the download links of the software.

3 Installation

See the [BASIS guide on software installation](#) for a complete list of build tools and detailed installation instructions.

3.1 Prerequisites

Dependency	Version*	Description
BASIS	2.1.2	A meta-project developed at SBIA to standardize the software development.
DRAMMS	1.4.1	A registration algorithm developed at SBIA to warp images.
AFNI		Using the version built on 2008_07_18_1710
FSL	4.1.5	A comprehensive library of analysis tools for brain imaging data
SCIKIT-LEARN	0.14.1	A python package providing several data mining and data analysis tools.
NIBABEL	1.2.0	A python package for read and write access to common medical file formats

* The versions listed are the minimum versions of the softwares for which the MASS package was tested.

3.2 Job Scheduler

If you have access to a computing cluster which has a job scheduler/queuing software (SGE, PBS etc) installed, it can be used to significantly reduce the (wall-clock) time it will take for the MASS software to produce the results. During the installation process, you can initialize the `SCHEDULER` variable with the particular version of your job scheduler. Currently, there are four options that are supported. You can select the one that best fits your system:

```
SGE - Sun Grid Engine
PBS - Portable Batch System
NONE - No queuing system (default)
MISC - User defined setting
```

If you have a different queuing software and you select the “MISC” option, you need to modify the `src/schedulerSettings/SettingsMISC.sh` file within the package with the appropriate options and arguments that are specific to your queuing system. You can refer to the corresponding files for SGE and PBS as examples.

3.3 Configure

1. Extract source files:

```
tar -xzf mass-1.1.0-source.tar.gz
```

2. Create build directory:

```
mkdir mass-1.1.0-build
```

3. Change to build directory:

```
cd mass-1.1.0-build
```

4. Run `CMake` to configure the build tree by using either one of the following commands:

```
cmake -D CMAKE_INSTALL_PREFIX:STRING=/Full/path/to/install/mass/
-D SCHEDULER:STRING=??? ../mass-1.1.0-source
```

OR:

```
ccmake ../mass-1.1.0-source
```

- Press `c` to configure the build system and `e` to ignore warnings.
- Set `SCHEDULER` variable with your job scheduler information.
- Set `CMAKE_INSTALL_PREFIX` and other CMake variables and options.
- Continue pressing `c` until the option `g` is available.
- Then press `g` to generate the GNU Make configuration files.

3.4 Build

After the configuration of the build tree, the software can be built using GNU Make:

```
make
```

3.5 Test

After building the software, the software tests can be run using

```
make test
```

Allow 30-60 mins for the tests to finish. The last test, if the `SCHEDULER` variable is not set to `NONE`, is meant to check if submitting the jobs to the queuing system works. Please check your queue (for e.g. using `qstat` for SGE, PBS) to make sure that the jobs were submitted. If they are submitted, you can either delete them or wait for them to finish. As soon as these tests finish, you can proceed to the installation.

3.6 Install

The final installation copies the built files and additional data and documentation files to the installation directory specified using the `CMAKE_INSTALL_PREFIX` option during the configuration of the build tree:

```
make install
```

After the successful installation, the build directory can be removed again.

4 Manual

4.1 General Processing Pipeline

- Step 1.** Bias correct (N3, N4 or any other method) all of the images so that any large scale inhomogeneities are removed and the input is of reasonable quality
- Step 2.** Run `ChooseTemplates` to select a set of subjects (7-20) to be used as templates for your entire study. If your study is large with a lot of variation (multi site, multi protocol etc), use more templates.
- Step 3.** Run `mass` on the bias corrected images selected in **Step 2** to skullstrip them using the generic templates provided along with the package. These subjects would be your gold standard for the rest of your study, so make sure the quality of the results on these subjects is good. You can manually correct these subjects if you choose to.
- Step 4.** Rename these results in the following format, with N being the template number from 1 to N :
- TemplateN.nii.gz* - The original intensity image.
- TemplateN_str_cbq.nii.gz* - The brain mask.
- Step 5.** Run `mass` on all subjects using these new templates by specifying the `-ref` option to use the newly created study-specific templates.
- Step 6.** Alternatively, you can skip steps 2-4 and use the generic templates on all of your subjects directly!

4.2 Template Selection

The template selection command within MASS which selects the cluster-centers within the dataset is named `ChooseTemplates`. The simplest use is:

```
ChooseTemplates -list /path/to/list/of/images.lst
```

This command will return a set of 6 images from the list that are the centroids of different clusters ($k=6$). Internally, it first resamples all of the input images to voxel dimensions 2:2:2 to enhance the processing speed and also reducing the memory requirement of the process. It then affinely registers all of the input images to the first image within the input list. So if you have a lot of images within the list, it may take a some time for it to process those images. In the last step of this process, it runs the `choosetemplates` executable which loads all the affinely registered images into memory and selects the cluster-centers.

If you have access to the computing cluster, it is **highly recommended** that you affinely register (using `flirt` with 12 dof) all of the input images to one of the randomly chosen images yourself and use the `-a` option as shown below to notify the script that the input images are already affinely registered. This will reduce the processing time significantly.

Supported File Formats: NIFTI-1 (recommended)

Supported Datatypes: byte (unsigned char, uint8), int8, short, int16, uint16, float, float32, int32.

Alternatively, the number of desired clusters can be specified using the `-clust` option as:

```
ChooseTemplates -list /path/to/list/of/images.lst -clust 15
```

If you would like to speed up the processing within this command, you can use multiple CPU cores on the cluster or on your computer using the `-MT` option :

```
ChooseTemplates -list /path/to/list/of/images.lst -clust 15 -MT 6
```

If you have already affinely registered all input images so they are in the same space, you can specify that by using the `-a` option. Here, you do not need the `-MT` option as all images have already been registered :

```
ChooseTemplates -list /path/to/list/of/images.lst -clust 15 -a
```

Additionally, you can also submit this script to your computing cluster if you have a large number of images that need be clustered. Make sure that you use the appropriate options for requesting memory as well as the number of threads while submitting this script cluster. This will ensure that the script does not run out of memory or overload the cluster.

4.3 MASS Default Command

The main command of MASS which removes the skull and other non-brain tissues is named `mass`. The simplest use is:

```
mass -in /path/to/source/sourceimage.hdr
```

This command will generate 3 files in the `/path/to/source/` directory:

sourceimage_brain.nii.gz - The skull-stripped image

sourceimage_brainmask.nii.gz - The final brain mask

sourceimage_brain_JacRank.nii.gz - The Jacobian Rank Mask

The Jacobian Rank Mask is a combination of the different registrations weighted locally by their jacobian determinants. This file can be thresholded using `mass-thresholdjacobian` to create stricter or more lenient masks. The maximum value within this file can vary with the number of templates that are used for registrations. Since it is a sum of ranks, that max value will always be $N * (N + 1) / 2$, where N is the number of templates used.

Supported File Formats: NIFTI-1 (recommended)

Supported Datatypes: byte (unsigned char, uint8), int8, short, int16, uint16, float, float32, int32.

4.4 MASS Options

To run the `mass` script which will internally do the processing and then submit the `mass-registrations` and `mass-skullStripping` jobs:

```
mass
-in /Path/To/Source/Directory/Input_n3.nii.gz
-dest /Path/To/Destination/Directory/;
```

To use the templates without the cerebellum:

```
mass
-in /Path/To/Source/Directory/Input_n3.nii.gz
-dest /Path/To/Destination/Directory/
-noCere;
```

To use a user defined set of templates:

```
mass
-in /Path/To/Source/Directory/Input_n3.nii.gz
-dest /Path/To/Destination/Directory/
-ref /Path/To/User/Defined/Templates/;
```

To remove the skull from the input image using the default options. However, do not use the computing cluster but run the `mass-registrations` jobs serially :

```
mass
-in /Path/To/Source/Directory/Input_n3.nii.gz
```

```
-dest /Path/To/Destination/Directory/  
-NOQ;
```

To remove the skull from the input image using the default options, but without the computing cluster. Additionally, use 6 CPU cores during `mass-registrations` to speed up the process:

```
mass  
-in /Path/To/Source/Directory/Input_n3.nii.gz  
-dest /Path/To/Destination/Directory/  
-NOQ  
-MT 6;
```

To remove the skull from the input image that is larger than normal and therefore, needs more memory, request 20GB instead of the default 16GB:

```
mass  
-in /Path/To/Source/Directory/Input_n3.nii.gz  
-dest /Path/To/Destination/Directory/  
-mem 20;
```

4.5 Threshold Jacobian Rank Mask

By default, the Jacobian Rank Mask is thresholded at 50% of the max value and then processed to get the final binary brain mask. If you'd like to threshold the Jacobian Rank Mask at a different percent value, say 70% to make the output brain mask stricter than the default value, use the following command:

```
mass-thresholdJacobian  
-in /Path/To/Source/Directory/Input_n3.nii.gz  
-jacRank /Path/To/Source/Directory/Input_n3_cbq_JacobianRankMask.nii.gz  
-perThresh 70
```

On the other hand, if you want to threshold using an absolute value of the Jacobian Rank Mask, say 47, you can run:

```
mass-thresholdJacobian  
-in /Path/To/Source/Directory/Input_n3.nii.gz  
-jacRank /Path/To/Source/Directory/Input_n3_cbq_JacobianRankMask.nii.gz  
-absThresh 47
```

5 Publications

5.1 Methodology

[ARAD2013]

[MedIA2011]

[IPMI2009]

5.2 Validation

[WBIR2012]

6 People

6.1 Advisors

- Christos Davatzikos (Christos.Davatzikos@uphs.upenn.edu)

6.2 Software Development

- Jimit Doshi (Jimit.Doshi@uphs.upenn.edu)

6.3 Contributors

- Guray Erus (Guray.Erus@uphs.upenn.edu)
- Yangming Ou (Yangming.Ou@uphs.upenn.edu)
- Meng-Kang Hsieh (Meng-Kang.Hsieh@uphs.upenn.edu)
- Bilwaj Gaonkar (Bilwaj.Gaonkar@uphs.upenn.edu)

6.4 Testers

- Harsha Battapady
- Xiao Da
- Meng-Kang Hsieh
- Guray Erus
- Martin Rozycki

References

- [ARAD2013] Jimit Doshi, Guray Erus, Yangming Ou, Bilwaj Gaonkar, Christos Davatzikos. Multi-Atlas Skull-Stripping, MASS. Academic Radiology ([web](#)).
- [MedIA2011] Yangming Ou, Aristeidis Sotiras, Nikos Paragios, Christos Davatzikos. DRAMMS: Deformable registration via attribute matching and mutual-saliency weighting. Medical Image Analysis 15(4): 622-639 (2011). ([web](#))
- [IPMI2009] Yangming Ou, Christos Davatzikos. DRAMMS: Deformable Registration via Attribute Matching and Mutual-Saliency Weighting. IPMI 2009: 50-62. ([web](#))
- [WBIR2012] Yangming Ou, Dong Hye Ye, Kilian M. Pohl, Christos Davatzikos. Validation of DRAMMS among 12 Popular Methods in Cross-Subject Cardiac MRI Registration. WBIR 2012: 209-219. ([web](#))