

# Wisconsin White Matter Hyperintensities Segmentation Toolbox

Wisconsin Alzheimer's Disease Research Center, UW Madison - v1.3

November 4, 2013

## W2MHS - About

**W2MHS** is an open source toolbox designed for detecting and quantifying White Matter Hyperintensities (WMH) in Alzheimer's and aging related neurological disorders. WMHs arise as bright regions on T2-weighted FLAIR images. They reflect comorbid neural injury or cerebral vascular disease burden. Their precise detection is of interest in Alzheimer's disease (AD) with regard to its prognosis. Our toolbox provides self-sufficient set of tools for segmenting these WMHs reliably and further quantifying their burden for down-processing studies. This documentation provides the background on the algorithm and parameters that comprise **W2MHS** along with the syntax.

## Package Details

**W2MHS** is designed in MATLAB. It uses Image Processing and Image Analysis Toolboxes (apart from the basic MATLAB commands). For details about the MATLAB software (installation, support and licensing), please go to MATLAB. The inputs to the toolbox include a T1 weighted MRI image and a T2 FLAIR image. Hence W2MHS uses multichannel images to segment WMHs. The several modules of **W2MHS** include:

- *Pre-processing Module*: Pre-processing involves constructing the White Matter (WM) region of interest and partial volume estimates of the tissues (White Matter, Gray Matter, Cerebro Spinal Fluid) among other things (bias correction, registration etc.). **W2MHS** uses a popular neuroimaging toolkit (SPM12b) for this pre-processing.
- *Segmentation Module*: Segmentation module is the heart of **W2MHS**. Random forest based regression method is used to detect the WMHs.
- *Quantification Module*: The purpose of this module is to summarize the WMH segmentations. This is especially important in down-stream analysis.

We first walk you through installation if the toolbox, followed by the several ways to interface with it. Finally there are details on some more advanced features and parameters of W2MHS.

## Installation

First, unpack the zipfile into the MATLAB toolbox or a directory of your choosing. As described above, W2MHS uses the latest version of SPM 12 beta for preprocessing which can be accessed here. Open MATLAB from the W2MHS directory and run the script **installW2MHS.m** which will compile the random forest and region growing mex files and add W2MHS to the MATLAB search path. Make sure to have a C++ compiler setup with MatLab. To do this enter '**mex -setup**' on the command line and select your compiler.

## Usage

### Graphical Interface

Enter "W2MHS" on the command line or run the "W2MHS.m" script to open up a GUI window. To get started check if the W2MHS Toolbox Path is not automatically entered; if not, correct that field. Next, make sure the address of SPM12b is entered correctly in the "SPM Toolbox Path" field.

You can go to File : Set Default on the menu bar to set the default configuration of the GUI. You may want the GUI to store both the SPM12b and W2MHS toolbox paths. Setting a new default will overwrite the old configuration. Note that you can also save sessions to '.mat' files and restore them later.

Make sure to enter all fields before running on subjects. Enter a name and output path, then you may add subjects using the "Add T1 and T2 Volumes" button. To remove a subject from the batch simply select it and press the "Remove Selected" button.

The add subjects button will allow you to add one subject at a time or select a directory that contains a batch of T1 and T2 image pairs. To use the directory batch function, the images must follow the naming convention described in the '**BatchSetup.m**' script. Each subject must have a unique name or identifier which is the Subject ID. If you enter one subject at a time make sure to enter this field and make sure they are all unique. When using the directory batch function make sure that the subjects it imports are automatically assigned Subject IDs.

## Main Script Parameters

Syntax of the W2MHS main script for command line usage.

**WhyD\_setup** (output\_name, output\_path, input\_images, output\_ids, w2mhstoolbox\_path, spmtoolbox\_path, do\_train, do\_preproc, do\_quantify)

1. output\_name : Name for the experiment (takes in only one field)
2. output\_path : Path of the directory where the outputs are to be stored (takes in only one field)
3. input\_images : Path of the T1 and T2 FLAIR images (.nii files) (number of fields is 2 times the number of subjects)
4. output\_ids : IDs for the subjects (number of fields is the same as the number of subjects)
5. w2mhstoolbox\_path : Path of W2MHS
6. spmtoolbox\_path : Path of SPM12b
7. do\_train : Optional argument for training (default : no)
8. do\_preproc : Optional argument for preprocessing (default : yes)
9. do\_quant : Optional argument for quantification (default : yes)

## Command Line Example

To better understand the syntax and usage of **WhyD\_setup.m**, here is an example run. Given he and T2-FLAIR images from 3 subjects and we want detect and quantify the WMH volume in each. The name for these experiments is "mystudy". We want to store all the outputs in "/home/myname/myoutputs". We denote the "1", "2" and "3". And their T1 and T2-FLAIR .nii image files are at "T1ofsubject1", "T2ofsubject1", "T1ofsubject2", "T2ofsubject2", "T1ofsubject3", "T2ofsubject3" respectively. With this setup, we then generate a example.m file as follows, and run it:

```
%% function example
clear all; clc; close all;
output_path = {'/home/myname/myoutputs'};
output_name = {'mystudy'};
output_ids = {'1'; '2', '3'};
input_images = {'T1ofsubject1', 'T2ofsubject1' ; 'T1ofsubject2', 'T2ofsubject2' ; ...
'T1ofsubject3', 'T2ofsubject3'};
input_meth = {'rf_regress'};
w2mhstoolbox_path = '/home/.../W2MHS';
spmtoolbox_path = '/home/.../spm12b';
do_train = 'no'; do_preproc = 'no'; do_quantify = 'yes';
param(.6,2.5,'yes');
WhyD_setup(output_name, output_path, input_images, output_ids, w2mhstoolbox_path, ...
spmtoolbox_path, do_train, do_preproc, do_quantify);
```

The outputs of these experiments are stored in three directories for each of the three subjects with names “mystudy 1”, “mystudy 2” and “mystudy 3”. Their paths are “/home/myname/myoutputs/mystudy 1”, “/home/myname/myoutputs/mystudy 2” and “/home/myname/myoutputs/mystudy 3” respectively. In each of these directories the outputs include final WMH probability maps of WMH detections, WMH quantification measures, preprocessing files like GM, WM and CSF, ventricular maps etc. These folder contain a .mat file called “names\_id.mat” (where “id” is the subject id). It gives details about all the output files. For example, this mat file for our subject “1” looks something like:

```
directory_path: '/home/myname/myoutputs/mystudy_1'
  folder_name: 'mystudy'
  folder_id: '1'
  source_bravo: 'BRAVO_1.nii'
  source_flair: 'FLAIR_1.nii'
  flair_coreg: 'rFLAIR_1.nii'
  pve_flair_c1: 'c1rFLAIR_1.nii'
  pve_flair_c2: 'c2rFLAIR_1.nii'
  pve_flair_c3: 'c3rFLAIR_1.nii'
  bias_corr: 'mrFLAIR_1.nii'
  Vent_bin: 'Vent_bin_1.nii'
  WM_mod: 'WM_modstrip_1.nii'
  GMCSF: 'GMCSF_strip_1.nii'
  th: 974.0644
  seg_out: 'RFREG_out_1.nii'
  seg_unrectify: 'RFREG_unrectify_1.nii'
  seg_pmap: 'RFREG_pmap_1.nii'
```

Of particular importance here are the fields “seg\_pmap” and the file “RFREG\_ev\_ID.mat” which represent the final WMH image (probability map) and its effective volume (EV) quantification (.mat file).

## Batch Script

W2MHS v1.3 also comes with a batch script for processing a lot of subjects. The GUI has a similar batch feature but this batch script only requires a directory and it will figure out the rest. There is a particular file setup and naming convention that must be followed for any of the W2MHS batch scripts to read in your inputs. To learn more about the usage of this script read the comments in the **BatchSetup.m** script.

## Advanced Options

**Training** We have provided features as well as a learned RF Regression model with the W2MHS toolbox but you may use your own features as well. The raw features are not included in the source code download but may be downloaded separately. They are approximately 1GB. See the examples, 'features training.mat' paired with 'labels training.mat' in 'W2MHS/training' to get feel for the format of the features we use. Our current model uses 119,284 feature vectors, half of which are extracted from voxel patches identified as white matter hyperintensities and half from voxel patches that are not hyperintense. We are currently working on providing code that will assist our users in creating their own features to train a more individualized RF model.

**Hyperparameters** Most of the parameters for the detection and preprocessing are set to reasonable values. Three parameters of importance are “clean\_th” (a cleaning threshold parameter) and “pmap\_cut” (a probability map cut value that is used for hyperintensity accumulation) and a “Yes” / “No” option to conserve disk space (W2MHS will delete extraneous intermittent files throughout processing). Note that this option will use approximately one fifth of the disk space as the traditional output. To edit the default settings simply open the **param.m** script in the W2MHS source folder and modify the parameters. More details on what these parameters do are included in the **param.m** script.

It is also easy to modify these hyperparameters for individual runs. If you are using the GUI, you will

see the three options at the bottom. Simply modify them to meet your needs. If you are running W2MHS via the command line, before the WhyD\_Setup script is called you must call the **param.m** script. Specify the three hyperparameters in the arguments of this function call.

Example: `param( 0.6, 2.5, 'yes');`

The first is the “pmap\_cut” second is “clean\_th” and last is “Conserve Disk Space.” If you do not call this script, the default parameters will be used.

## Contents

### Scripts

- **WhyD\_setup.m** : This is the *main* setup script. It creates the necessary directories and calls other internal scripts.
- **WhyD\_batch.m** : An internal batch script. Use BatchSetup.m if you want to do a command line batch automatically.
- **WhyD\_preproc.m** : *Internal* preprocessing script that generates a SPM12 batch processing to coregister and segment WM, GM and CSF tissues. It also constructs ventricle maps and PV estimates.
- **WhyD\_detect.m** : *Internal* segmentation script that performs the hyperintensity segmentation using random forest algorithm.
- **WhyD\_postproc.m** : *Internal* postprocessing script that cleans up the segmented outputs.
- **WhyD\_quant.m** : *Internal* quantification script that calculates the hyperintensity accumulation (deep, periventricular and total).
- **WhyD\_train.m** : *Internal* training script. This is an *optional* script that generates a learned model using a given set of features and labels (i.e. training data). Is the default setting this script is not used (a pre-generated model is included in the toolbox)
- **getKernels.m, getCenter.m, get\_gauss\_conv.m** : *Internal* scripts called by **WhyD\_detect.m** and **WhyD\_train.m**.
- **check\_preproc.m, check\_training.m** : *Internal* scripts called by **WhyD\_setup.m**.
- **BatchSetup.m** : Easy way to run a batch. Open this script and edit the parameters inside.
- **W2MHS.m** : Opens the GUI interface
- **installW2MHS.m** : Adds W2MHS to your MatLab filepath and compiles all mex files. Run this script once when you install the package.
- **params.m** : Contains hyperparameters which may be modified and saved in the W2MHS directory.
- **NIFTI\_codes** : Folder containing NIFTI codes (for .nii files).
- **rf\_codes** : Folder containing Mex and .m files for random forest algorithm.
- **training** : Folder containing training data and extracted model.

## Notes

This toolbox was developed on Linux operating systems. It will work on Windows if you have the right compilers installed. To connect your .mex compatible compiler with Matlab enter the command ”mex -setup” and select your compiler from the list. If you are having trouble running W2MHS on Windows or if you have any other bug reports or feature requests, post in the help forums on NITRC or SourceForge for assistance.