HyperMapp3r Documentation

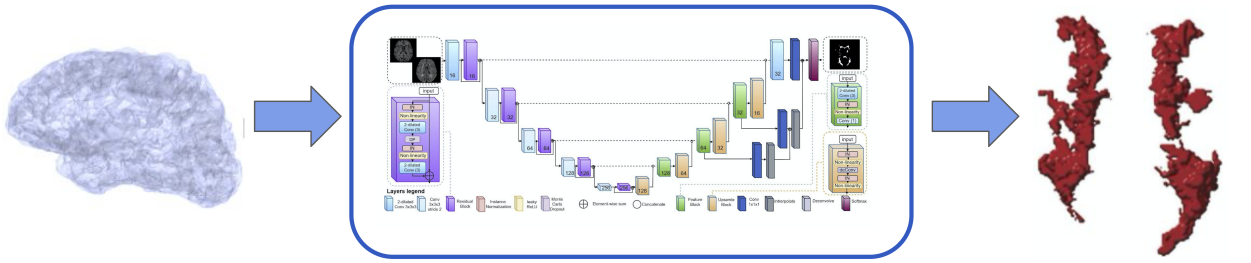
Maged Goubran, Parisa Mojiri, Mahdi Biparva, Edward Ntiri, Sand

Apr 24, 2024

CONTENTS:

HyperMapp3r (pronounced hypermapper) is a CNN-based segmentation algorithm of White Matter Hyperintensity(WMH) segmentation using MRI images from BrainLab. It can deal with brains with extensive atrophy and segments the wmh in seconds. It uses a T1-weighted, FLAIR, and brain mask images as the inputs and segments.

We recommend using HyperMapp3r with the Docker or Singularity containers we provide but it can also be installed locally. See our *installation instructions* for more information.



Inputs MRI:
T1, FLAIR and brain mask

HyperMapp3r

WMH segmentation

==========

CHAPTER

ONE

BEFORE INSTALLING HYPERMAPP3R

==========

1.1 Acknowledging this work

If you wish to include results generated by HyperMapp3r in a publication, please include a line such as the following:

- White Matter Hyperintensity (WMH) segmentation was performing using the HyperMapp3r algorithm (*hypermapp3r.readthedocs.io*) based on a convolutional neural network.

1.2 Reference

Mojiri P, Biparva M, Ntiri EE, Ramirez J, Boone L, Holmes M, Adamo S, Gao F, Ozzoude M, Scott C, Dowlatshahi D, Lawrence-Dewar J, Kwan D, Lang A, Marcotte K, Leonard C, Rochon E, Heyn C, Bartha R, Strother S, Tardif JC, Symons S, Masellis M, Swartz R, Moody A, Black SE*, Goubran M*. Deep Bayesian networks for uncertainty estimation and adversarial resistance of white matter hyperintensity segmentation. *Human Brain Mapping* 2022. doi: https://doi.org/10.1002/hbm.25784

1.3 License

==========

CHAPTER

TWO

LOCAL INSTALL

==========

Important:

`Docker` is our recommended method for running `HyperMapp3r` on local machines and servers. We recommend `Singularity` to run `HyperMapp3r` in a cluster environment (e.g. Compute Canada). For information on how to install and use these, please see Docker / Singularity

2.1 Python

For the main required Python packages (numpy, scipy, etc.) we recommend using Anaconda for Python 3.6

2.2 ANTs & Convert3D (c3d)

If either ANTs or c3d are not installed on your machine, run `install_depends.sh`, located in the project directory. The required software will be installed in the `depends` directory. If you are intrested to install c3d on MacOS or Windows, you can dowanlod it from this link.

2.3 Installing package and dependencies for HippMapp3r locally

1. Clone repository

```
git clone https://github.com/AICONSlab/HyperMapp3r.git HyperMapp3r

(or install zip file and uncompress)

cd HyperMapp3r
```

If you want to create a virtual environment where HyperMapp3r can be run,

```
conda create -n hypermapper python=3.6 anaconda
source activate hypermapper
```

To end the session, deactivate the environment

```
source deactivate
```

To delete the environment,

```
conda env remove --name hypermapper
```

2. Install dependencies

```
pip install git+https://www.github.com/keras-team/keras-contrib.git
```

If the computer you are using has a GPU:

```
pip install -e .[hypermapper_gpu]
```

If not:

```
pip install -e .[hypermapper]
```

3. Test the installation by running

```
hypermapper --help
```

To confirm that the command line function works, and

```
hypermapper
```

To launch the interactive GUI.

2.4 Download deep models

Download the models from this link and place them in the `models` directory

2.5 For tab completion

```
pip3 install argcomplete
activate-global-python-argcomplete
```

2.6 Updating HyperMapp3r

To update HyperMapp3r, navigate to the directory where HyperMapp3r was cloned and run

```
git pull
pip install -e .[{option}] -process-dependency-links
```

where "option" is dependent on whether or not you have a GPU (see package installation steps above)

==========

CHAPTER

THREE

GETTING STARTED

==========

You can use HyperMapp3r through the graphical user interface (GUI) or command line:

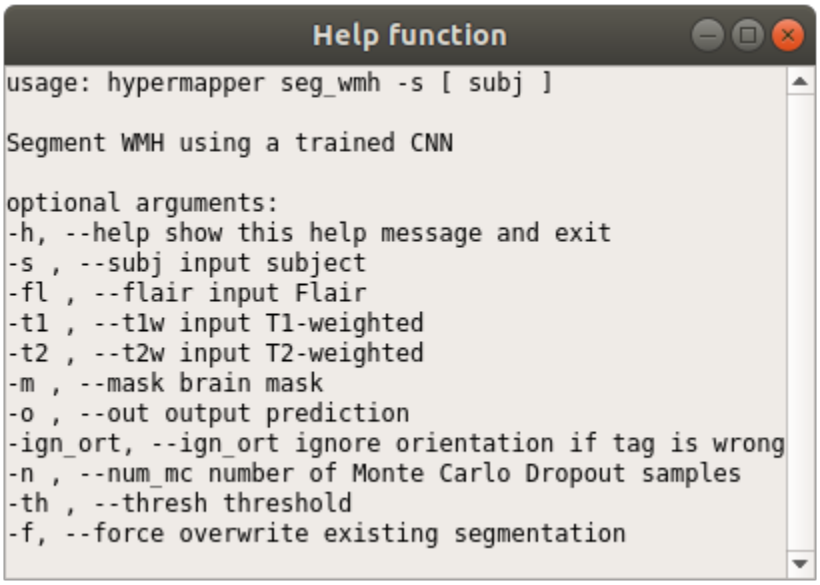
3.1 For GUI

To start the GUI, type

```
hypermapper
```

A GUI that looks like the image below should appear. You can hover any of buttons in the GUI to see a brief description of the command.

You can get the command usage info by clicking the “Help” box on any of the pop-up windows.



3.2 For Command Line

You can see all the hypermapper commands by typing either of the following lines:

```
hypermapper -h
hypermapper --help
```

Once you know the command you want to know from the list, you can see more information about the command. For example, to learn more about seg_wmh:

```
hypermapper seg_wmh -h
hypermapper seg_wmh --help
```

3.3 WMH volumes

To extract WMH volumes use the GUI (Stats/WMH) or command line:

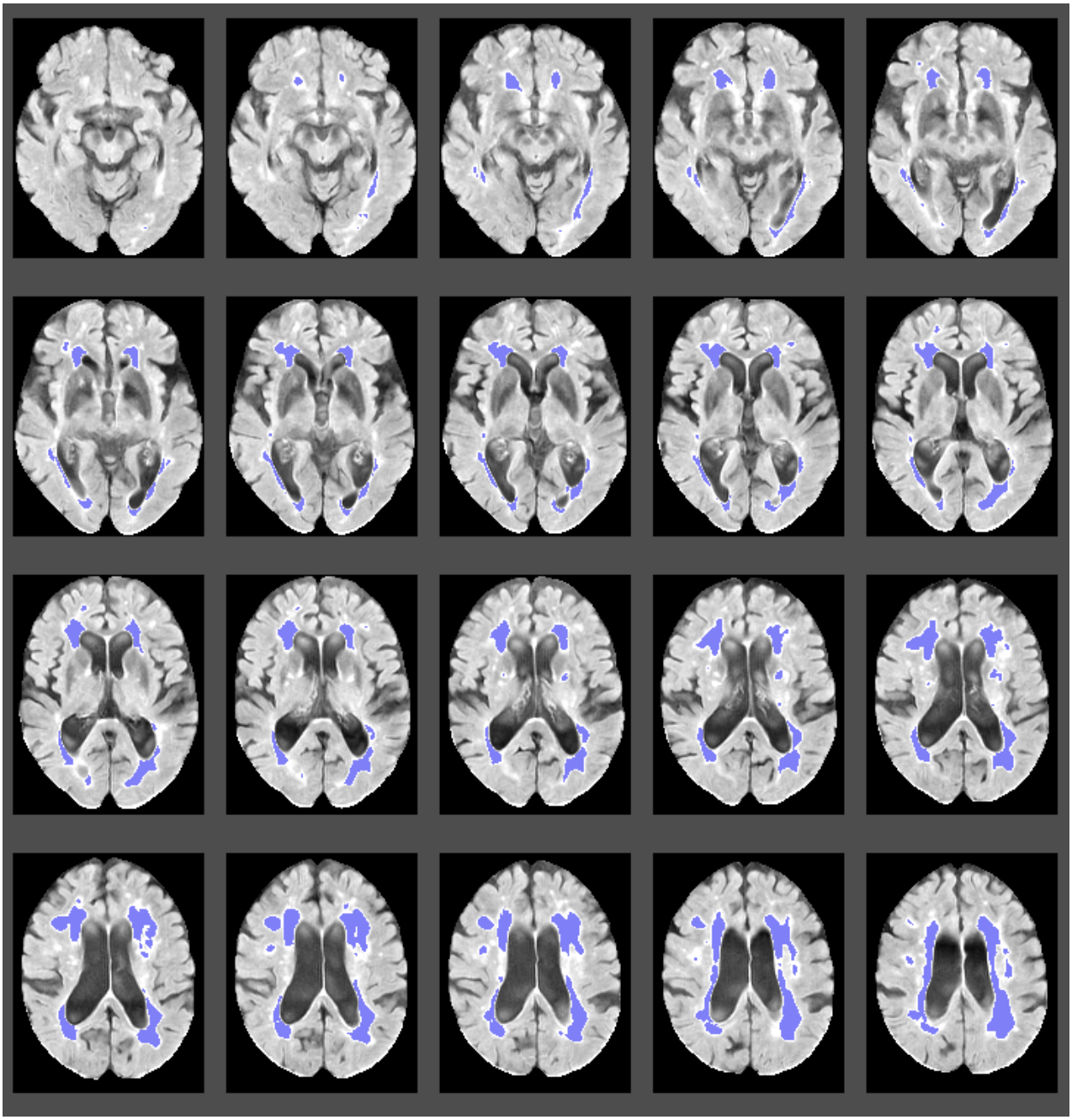
```
hypermapper stats_wmh -h
```

3.4 QC

QC files are automatically generated in a sub-folder within the subject folder. They are .png images that show a series of slices in the brain to help you quickly evaluate if your command worked successfully, especially if you have run multiple subjects. They can also be created through the GUI or command line:

```
hypermapper seg_qc -h
```

The QC image should look like this:



3.5 Logs

Log files are automatically generated in a sub-folder within the subject folder. They are .txt files that contain information regarding the command and can be useful if something did not work successfully.

3.6 File conversion

Convert Analyze to Nifti (or vice versa)

```
hypermapper filetype

Required arguments:
-i , --in_img    input image, ex:MM.img
-o , --out_img   output image, ex:MM.nii

Example:
hypermapper filetype --in_img subject_T1.img --out_img subject_T1.nii.gz
```

##########################

CHAPTER

FOUR

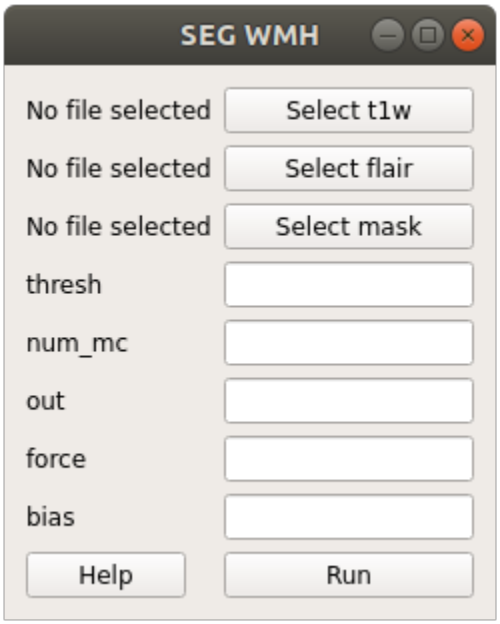
SEGMENTATION TUTORIALS

##########

4.1 GUI

After opening the HyperMapper GUI, click “WMH” under the “Segmentation” tab. Wait for a new pop-up window to appear.

Click “Select t1w” and chose your T1 image.



Type your desired output name in the "out" box. Click "Run". Your output file will automatically appear in your t1w folder.

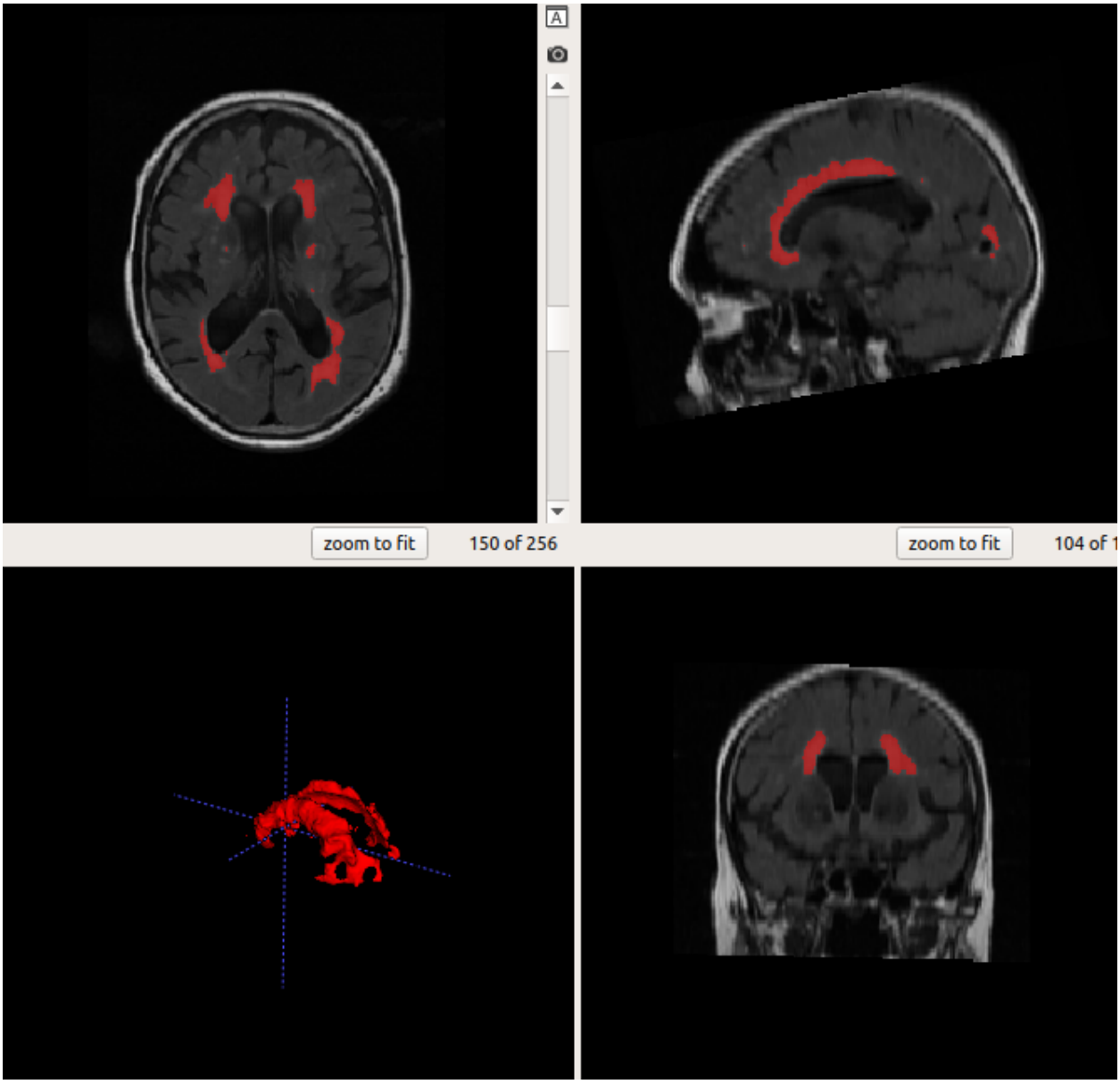
4.2 Command Line

```
hypermapper seg_wmh

optional arguments:
-h, --help            show this help message and exit
-s , --subj           input subject
-fl dir, --flair dir  input Flair
-t1 , --t1w           input T1-weighted
-m , --mask           brain mask
-o , --out            output prediction
-ign_ort, --ign_ort   ignore orientation if tag is wrong
-n , --num_mc         number of Monte Carlo Dropout samples
-th , --thresh        threshold
-f, --force           overwrite existing segmentation

Examples:
hypermapper seg_wmh -s subjectname -b
hypermapper seg_wmh -t1 subject_T1_nu.nii.gz -fl subject_T1acq_nu_FL.nii.gz -m subject_
↪T1acq_nu_HfB.nii.gz -o subject_wmh_pred.nii.gz
```

The output should look like this.:


zoom to fit
150 of 256
zoom to fit
104 of 1

==========

CHAPTER

FIVE

DOCKER / SINGULARITY

==========

If you intend to use Singularity, scroll down to the Singularity section. Otherwise, the steps to use the image in Docker can be found below.

5.1 Before using Docker image for HyperMapp3r

If you want to use Docker to run HyperMapp3r, you must first install Docker on your system. While the installation method differs per system, instructions can be found for the following:

- Ubuntu
- Windows
- Mac

Once Docker is installed, open the docker terminal and test it with the command

```
docker run hello-world
```

5.2 Pulling HyperMapp3r's Docker image

While you can download various Docker images, for the sake of this tutorial pull the HyperMpp3r image

```
docker pull mgoubran/hypermapper:latest
```

Verify that the image was pulled successfully by checking all images on your system

```
docker images
```

5.3 Running the Docker image

If you have installed Docker for the first time and have verified that the `hello-world` image was running, then HyperMapper can be run on your syste.

The simplest way to run the container is:

```
docker run -it mgoubran/hypermapper seg_wmh -t1 /hypermapper/data/test_case/t1.nii.gz -
↪fl /hypermapper/data/test_case/fl.nii.gz -m /hypermapper/data/test_case/mask.nii.gz
```

To run the Docker container in an interactive shell, run

```
docker run --rm -v {enter/path/here}:/root -it --entrypoint /bin/bash mgoubran/
↪hypermapper
```

5.4 Using HyperMapper on Singularity

Docker images can still be used on Singularity. This is especially good if you are processing images using Compute Canada clusters. The following instructions are based on the steps provided on the Compute Canada wiki.

Load the specific Singularity module you would like to use.

```
module load singularity/3.5
```

Although hypermapper is stored as a Docker image, it can be built in singularity by calling:

```
singularity build hypermapper.sif docker://mgoubran/hypermapper
```

To ensure that the Docker image has been built in Singularity, run

```
singularity exec hypermapper.sif hypermapper --help
```