CONTENTS:

1 Indices and tables 5
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.

Copyright (C) 2020 BrainLab.

# Before installing HyperMapp3r

## 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](hypermapp3r.readthedocs.io)) based on a convolutional neural network.

## Reference

• Paper has been submitted to Human Brain Mapping journal (under review with minor revisions) and is available on [bioRxiv](https://www.biorxiv.org/content/10.1101/2021.08.18.456666v3).


HyperMapp3r is licensed under the terms of the GNU General Public License v3.0.

HyperMapp3r is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. You should have received a copy of GNU General Public License v3.0 along with HyperMapp3r.

The code is released for academic research use only. For commercial use, please contact [maged.goubran@sri.utoronto.ca](mailto:maged.goubran@sri.utoronto.ca).

# Local Install

## Python
For the main required Python packages (numpy, scipy, etc.) we recommend using [Anaconda for Python 3.6](https://www.continuum.io/downloads).

## 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 installeed in the `depends` directory. If you are intrested to install c3d on MacOS or Windows, you can dowanlod it from [this link](http://www.itksnap.org/pmwiki/pmwiki.php?n=Downloads.C3D).

## Installing package and dependencies for HippMapp3r locally

1. Clone repository

```bash
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.

## Download deep models

Download the models from [this link](https://drive.google.com/drive/folders/1QS3t01jMSJq6zAftMDu1AzufplGjLODR) and place them in the models directory

## For tab completion

pip3 install argcomplete activate-global-python-argcomplete

## 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)

# Getting started

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

## 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.

![HyperMapp3r GUI](images/hypermapper_gui.png)

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

![HyperMapp3r Help](images/hypermapper_help.png)

## 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
```

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

```
hypermapper stats_wmh -h
```

## QC files
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:

![images/wmh_seg_qc.png](images/wmh_seg_qc.png)

## 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.

## 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

# Issues

# 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.

## Before using Docker image for HyperMap3r

If you want to use Docker to run HyperMap3r, you must first install Docker on your system. While the installation method differs per system, instructions can be found for the following:

- [Ubuntu](https://docs.docker.com/install/linux/docker-ce/ubuntu/)
- [Windows](https://docs.docker.com/docker-for-windows/install/)
- [Mac](https://docs.docker.com/docker-for-mac/)

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

```
docker run hello-world
```

## Pulling HyperMap3r’s Docker image

While you can download various Docker images, for the sake of this tutorial pull the HyperMap3r image

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

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

```
docker images
```
## 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 system.

The simplest way to run the container is:

```bash
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

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

## 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](https://docs.computecanada.ca/wiki/Singularity).

Load the specific Singularity module you would like to use.

```bash
module load singularity/3.5
```

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

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

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

```bash
singularity exec hypermapper.sif hypermapper --help
```
INDICES AND TABLES

- genindex
- modindex
- search