

---

# HippMapp3r Documentation

**Maged Goubran, Edward Ntiri, Melissa Holmes, Hassan Akhavein,**

**Apr 02, 2024**



## CONTENTS:

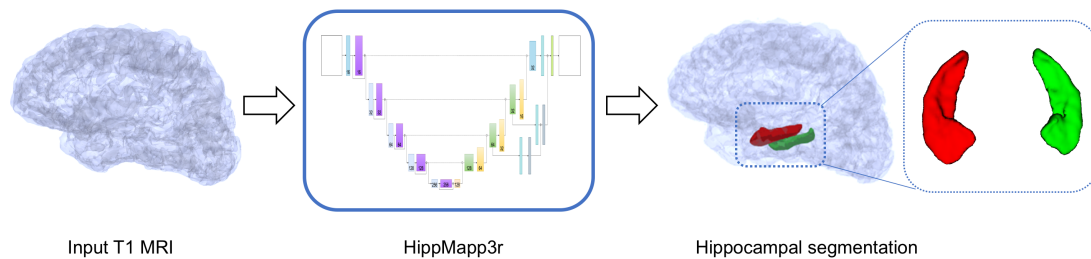
<b>1</b>	<b>Before installing HippMapp3r</b>	<b>3</b>
1.1	Acknowledging this work . . . . .	3
1.2	Reference . . . . .	3
1.3	License . . . . .	3
<b>2</b>	<b>Local Install</b>	<b>5</b>
2.1	Python . . . . .	5
2.2	ANTs & Convert3D (c3d) . . . . .	5
2.3	Installing package and dependencies for HippMapp3r locally . . . . .	5
2.4	Download deep models . . . . .	6
2.5	For tab completion . . . . .	6
2.6	Updating HippMapp3r . . . . .	6
<b>3</b>	<b>Getting started</b>	<b>7</b>
3.1	For GUI . . . . .	7
3.2	For Command Line . . . . .	9
3.3	Hippocampal volumes . . . . .	9
3.4	QC . . . . .	10
3.5	Logs . . . . .	10
3.6	File conversion . . . . .	10
<b>4</b>	<b>Segmentation tutorials</b>	<b>11</b>
4.1	GUI . . . . .	11
4.2	Command Line . . . . .	11
<b>5</b>	<b>Issues</b>	<b>13</b>
<b>6</b>	<b>Docker / Singularity</b>	<b>15</b>
6.1	Before using Docker image for HippMapp3r . . . . .	15
6.2	Pulling HippMapp3r's Docker image . . . . .	15
6.3	Running the Docker image . . . . .	15
6.4	Using HippMapper on Singularity . . . . .	16





*HippMapp3r* (pronounced hippmapper) is a CNN-based segmentation technique of the whole hippocampus using MRI images from BrainLab. It can deal with brains with extensive atrophy and segments the hippocampi in seconds. It uses a T1-weighted image as the only input and segments skull-stripped images. Note: skull-stripping is a required preprocessing step if the skull is present.

Copyright (C) 2019 AICONSLab.





## BEFORE INSTALLING HIPPMAPP3R

### 1.1 Acknowledging this work

If you wish to include results generated by HippMapp3r in a publication, please include a line such as the following:

- Whole hippocampal segmentation was performing using the HippMapp3r algorithm ([hippmapp3r.readthedocs.io](http://hippmapp3r.readthedocs.io)) based on a convolutional neural network.

### 1.2 Reference

If you use HippMapp3r in your work please cite our paper:

Goubran, M., Ntiri E., Akhavein, H., Holmes, M., Nestor, S., Ramirez, J., Adamo, S., Gao, F., Ozzoude, M., Scott, C., Martel, A., Swardfager, W., Masellis, M., Swartz, R., MacIntosh B, and Black, SE. “Hippocampal segmentation for atrophied brains using three-dimensional convolutional neural networks”. *Human Brain Mapping*. 2020 Feb 1;41(2):291-308. <https://onlinelibrary.wiley.com/doi/full/10.1002/hbm.24811>

### 1.3 License

HippMapp3r is licensed under the terms of the *GNU General Public License v3.0*.

HippMapp3r 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 HippMapp3r.

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

### 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

### 2.3 Installing package and dependencies for HippMapp3r locally

1. Clone repository

```
git clone https://github.com/mgoubran/HippMapp3r.git HippMapp3r

(or install zip file and uncompress)

cd HippMapp3r
```

If you want to create a virtual environment where HippMapp3r can be run,

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

To end the session, deactivate the environment

```
source deactivate
```

To delete the environment,

```
conda env remove --name hippmapper
```

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 .[hippmapper_gpu]
```

If not:

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

3. Test the installation by running

```
hippmapper --help
```

To confirm that the command line function works, and

```
hippmapper
```

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 HippMapp3r

To update HippMapp3r, navigate to the directory where HippMapp3r 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 HippMapp3r through the graphical user interface (GUI) or command line:

### 3.1 For GUI

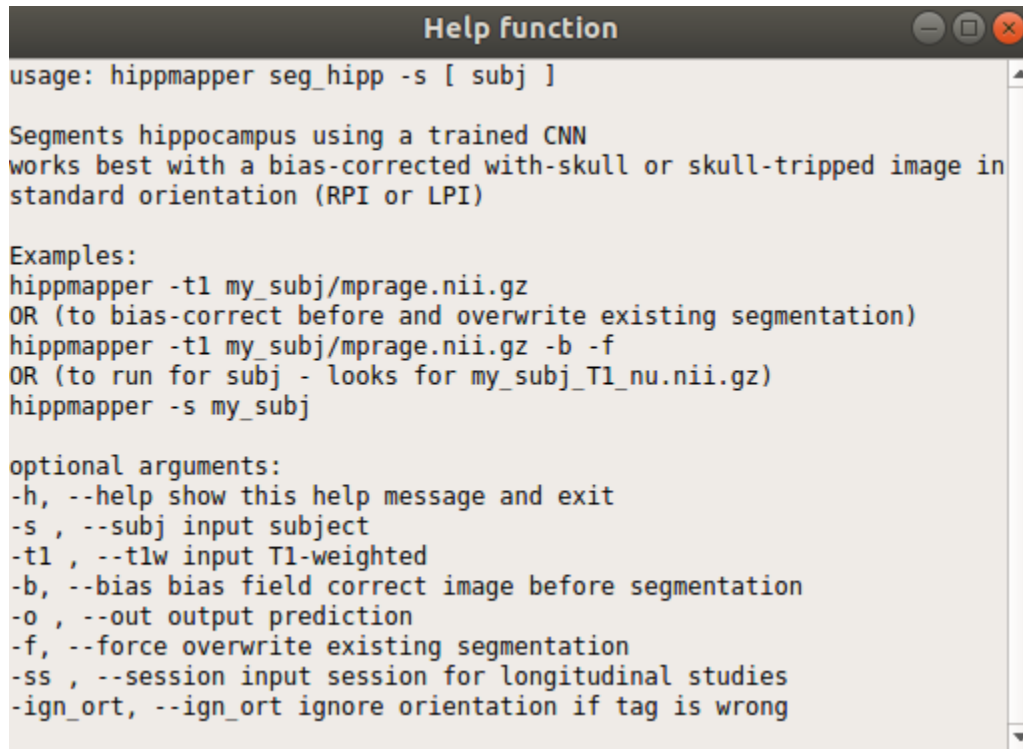
To start the GUI, type

```
hippmapper
```

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.



```

Help function
usage: hippmapper seg_hipp -s [ subj ]

Segments hippocampus using a trained CNN
works best with a bias-corrected with-skull or skull-tripped image in
standard orientation (RPI or LPI)

Examples:
hippmapper -tl my_subj/mprage.nii.gz
OR (to bias-correct before and overwrite existing segmentation)
hippmapper -tl my_subj/mprage.nii.gz -b -f
OR (to run for subj - looks for my_subj_T1_nu.nii.gz)
hippmapper -s my_subj

optional arguments:
-h, --help show this help message and exit
-s , --subj input subject
-tl , --tlw input T1-weighted
-b, --bias bias field correct image before segmentation
-o , --out output prediction
-f, --force overwrite existing segmentation
-ss , --session input session for longitudinal studies
-ign_ort, --ign_ort ignore orientation if tag is wrong

```

## 3.2 For Command Line

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

```

hippmapper -h
hippmapper --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\_hfb:

```

hippmapper seg_hipp -h
hippmapper seg_hipp --help

```

## 3.3 Hippocampal volumes

To extract hippocampal volumes use the GUI (Stats/Hippocampal Volumes) or command line:

```

hippmapper stats_hp -h

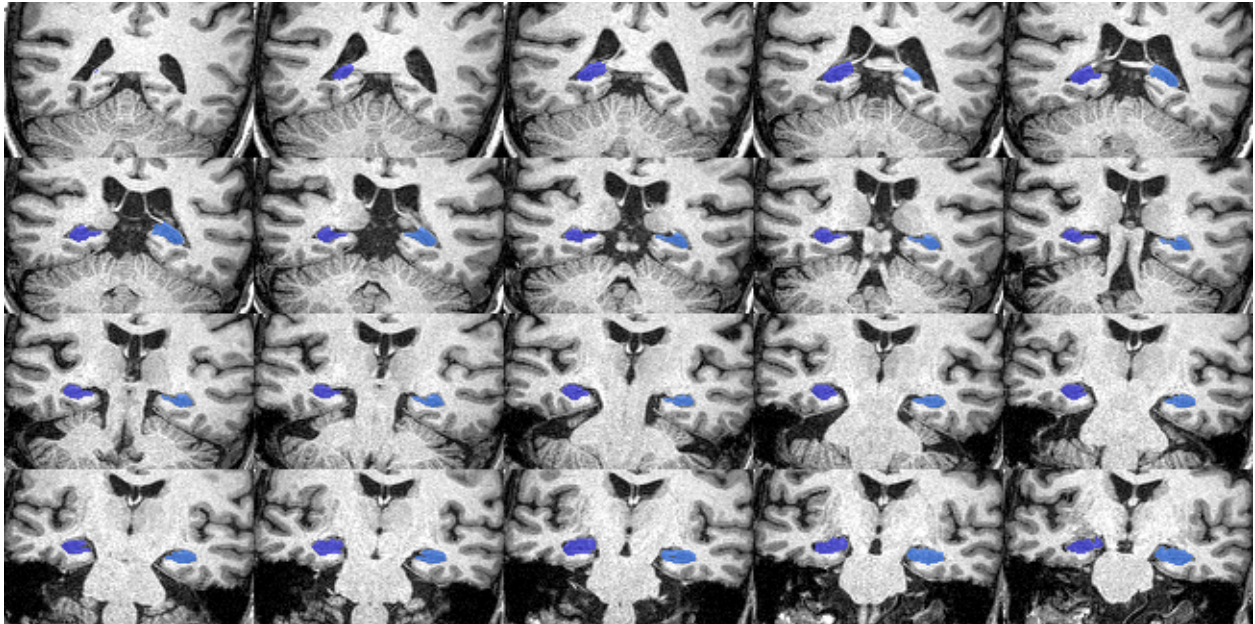
```

## 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:

```
hippmapper 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)

```
hippmapper filetype
```

Required arguments:

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

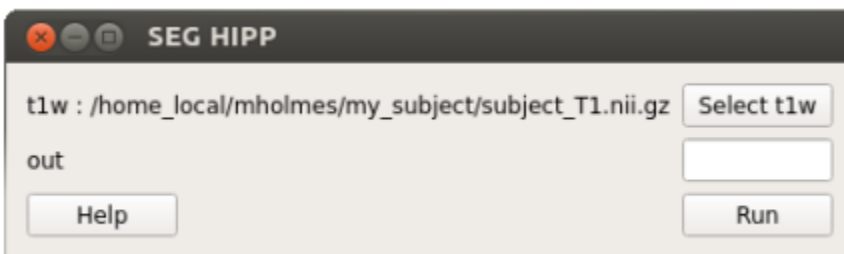
Example:

```
hippmapper filetype --in_img subject_T1.img --out_img subject_T1.nii.gz
```

## SEGMENTATION TUTORIALS

### 4.1 GUI

After opening the HippMapper GUI, click “Hippocampus” under the “Segmentation” tab. Wait for a new pop-up window to appear. The window should look like the image below.



Click “Select t1w” and chose your T1 image. Click “Run”. Type your desired output name in the “out” box. Your output file will automatically appear in your t1w folder.

### 4.2 Command Line

```
hippmapper seg_hipp
```

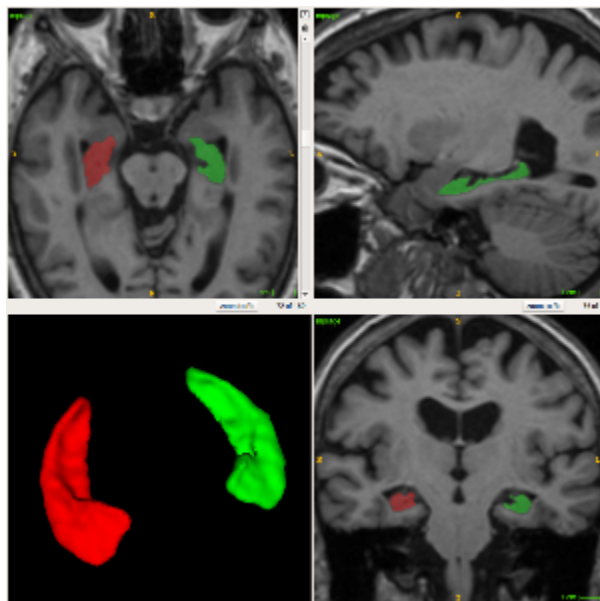
Optional arguments:

```
-s , --subj      input subject
-t1 , --t1w      input T1-weighted
-b , --bias      bias field correct image before segmentation
-o , --out       output prediction
-f , --force     overwrite existing segmentation
-ss , --session  input session for longitudinal studies
```

Examples:

```
hippmapper seg_hipp -s subjectname -b
hippmapper seg_hipp -t1 subject_T1_nu.nii.gz -o subject_hipp.nii.gz
```

The output should look like this.:





---

CHAPTER  
**FIVE**

---

**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.

### 6.1 Before using Docker image for HippMapp3r

If you want to use Docker to run HippMapp3r, 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
```

### 6.2 Pulling HippMapp3r's Docker image

While you can download various Docker images, for the sake of this tutorial pull the HippMpp3r image

```
docker pull mgoubran/hippmapper:latest
```

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

```
docker images
```

### 6.3 Running the Docker image

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

The simplest way to run the container is:

```
docker run -it mgoubran/hippmapper seg_hipp -t1 /hippMapp3r/data/test_case/mprage.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/hippmapper
```

### 6.4 Using HippMapper 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 HippMapper is stored as a Docker image, it can be built in singularity by calling:

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

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

```
singularity exec hippmapper.sif hippmapper --help
```