

WMH-SynthSeg

This module is currently only available in the development version of FreeSurfer

Author: Pablo Laso

E-mail: lasomielgopablo [at] gmail.com

Rather than directly contacting the author, please post your questions on this module to the FreeSurfer mailing list at freesurfer@nmr.mgh.harvard.edu

If you use WMH-SynthSeg in your analysis, please cite:

- [Quantifying white matter hyperintensity and brain volumes in heterogeneous clinical and low-field portable MRI](#). Laso P, Cerri S, Sorby-Adams A, Guo J, Matteen F, Goebel P, Wu J, Liu P, Li H, Young SI, Billot B, Puonti O, Sze G, Payabavash S, DeHavenon A, Sheth KN, Rosen MS, Kirsch J, Strisciuglio N, Wolterink JM, Eshaghi A, Barkhof F, Kimberly WT, and Iglesias JE. Proceedings of ISBI 2024 (in press).

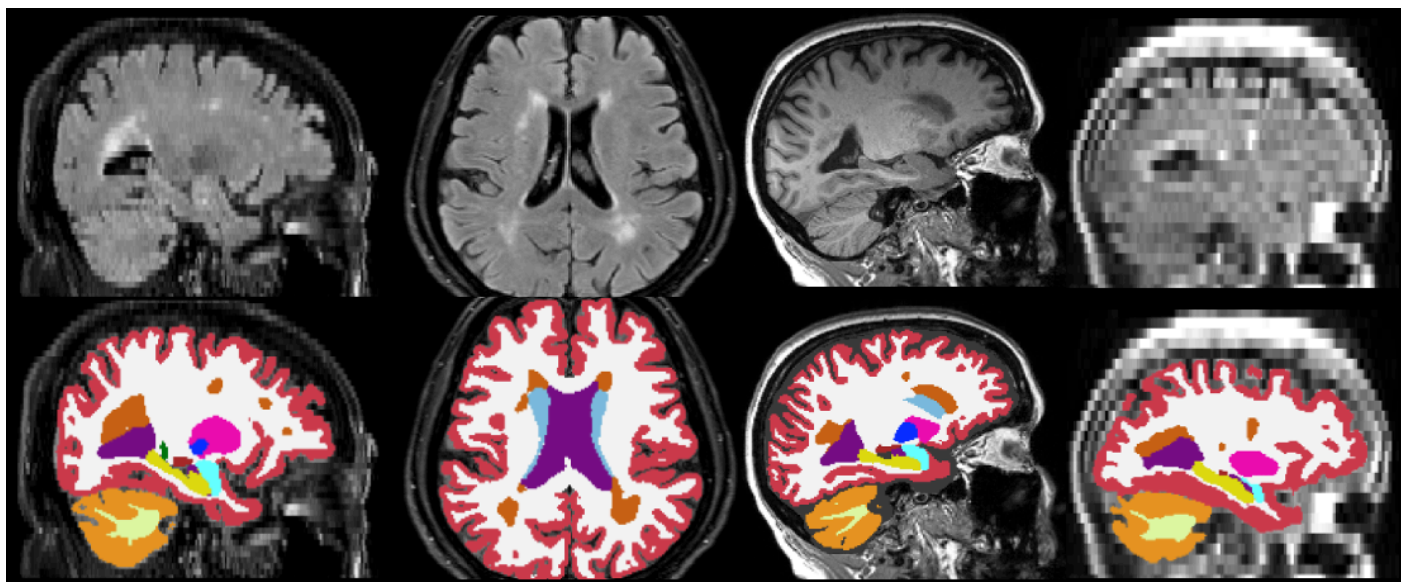
Contents

1. General Description
2. Usage
3. Frequently asked questions (FAQ)

1. General Description

This tool is a version of SynthSeg that, in addition to segmenting anatomy, also provides segmentations for white matter hyperintensity (WMH) - or hypointensities, in T1-like modalities. As the original SynthSeg, WMH-SynthSeg works out of the box and can handle brain MRI scans of any contrast and resolution. Unlike SynthSeg, WMH-SynthSeg is designed to adapt to low-field MRI scans with low resolution and signal-to-noise ratio (which makes it potentially a bit less accurate on high-resolution data acquired at high field).

As for SynthSeg, the output segmentations are returned at high resolution (1mm isotropic), regardless of the resolution of the input scans. The code can run on the GPU (3s per scan) as well as the CPU (1 minute per scan). The list of segmented structures is the same as for SynthSeg 2.0 (plus the WMH label, which is FreeSurfer label 77). Below are some examples of segmentations given by WMH-SynthSeg.



2. Usage

You can use WMH-SynthSeg with the following command:

```
mri_wmhsynthseg --i <input> --o <output> [--csv_vols <CSV file>] [--device <device>] [--threads <threads>] [--crop]
[--save_lesion_probabilities]
```

where:

- `<input>`: path to a scan to segment, or to a folder.
- `<output>`: path where the output segmentations will be saved. This must be the same type as `--i` (i.e., the path to a file or a folder).
- `<CSV file>`: (optional) path to a CSV file where volumes for all segmented regions will be saved.
- `<threads>`: (optional) number of threads to be used by PyTorch in CPU mode (see blow). Default is 1. Set it to -1 to use all possible cores.
- `<device>`: (optional) device used by PyTorch. Default is 'cpu'. Set it to 'cuda' to use the GPU, if available. If there are multiple GPUs available, use 'cuda:0', 'cuda:1', etc to index them.
- `<crop>`: (optional) runs 2 passes of the algorithm: one to roughly find the center of the brain, and another to do the segmentation processing only a portion of the image, cropped around the center. You will need to activate this option if you are using a GPU (long story).
- `<save_lesion_probabilities>`: (optional) saves additional files with the soft segmentations of the WMHs.

Important: If you wish to process several scans, we highly recommend that you put them in a single folder, rather than calling `mri_WMHsynthseg` individually on each scan. This will save the time required to set up the software for each scan.

3. Frequently asked questions (FAQ)

- **What are the computation requirements for this tool?**

About 32GB of RAM memory.

- **Does running this tool require preprocessing of the input scans?**

No! Because we applied aggressive augmentation during training (see paper), this tool is able to segment both processed and unprocessed data. So there is no need to apply bias field correction, skull stripping, or intensity normalization.

- **The sum of the number of voxels of a given structure multiplied by the volume of a voxel is not equal to the volume reported in the output volume file.**

This is because the volumes are computed upon a soft segmentation, rather than the discrete segmentation. The same happens with the main recon-all stream: if you compute volumes by counting voxels in `aseg.mgz`, you don't get the values reported in `aseg.stats`.

- **What formats are supported ?**

This tool can be run on Nifti (.nii/.nii.gz) and FreeSurfer (.mgz) scans.

- **How can I increase the speed of the CPU version without using a GPU?**

If you have a multi-core machine, you can increase the number of threads with the `--threads` flag (up to the number of cores).

- **Why are the inputs automatically resampled to 1mm resolution ?**

Simply because, in order to output segmentations at 1mm resolution, the network needs the input images to be at this particular resolution! We highlight that the resampling is performed internally to avoid the dependence on any external tool.

- **Why aren't the segmentations perfectly aligned with their corresponding images?**

This may happens with viewers other than FreeSurfer's Freeview, if they do not handle headers properly. We recommend using Freeview but, if you want to use another viewer, you may can use `mri_convert` with the `-rl` flag to obtain resampled images, which any other view will display correctly. Something like: `'mri_convert input.nii.gz input.resampled.nii.gz -rl segmentation.nii.gz'`.