Normalisation pipeline

Andreas G. Damsbo

2024-01-08

Table of contents

Preface **4**
Motivation . . . 4
License . . . 4
Glossary . . . 5

1 Contents **6**
1.1 Glossary . . . 6

2 Getting started **8**
2.1 Let's get to it . . . 8
2.2 Glossary . . . 10

3 Quality control **11**
3.1 Inspection . . . 11
3.2 Cropping . . . 11
3.3 Masking . . . 12
3.3.1 Too little . . . 12
3.3.2 Too much . . . 12
3.3.3 Register to MNI space again . . . 13
3.4 Final quality control steps . . . 13
3.5 Glossary . . . 14

I Use case #1: Finding NeMo **15**

4 Preparations **16**
4.1 Glossary . . . 16

5 Using the NeMo tool **17**
5.1 Package and upload . . . 17
5.2 Glossary . . . 17

6 ZIP to data set **18**
6.1 Visualisation . . . 19
6.2 Glossary . . . 21

Appendices **22**

A Changelog **22**
A.1 2024.1.1 . . . 22
A.2 2023.10.06 . . . 22
A.3 2023.10.05 . . . 23
A.3.1 Notes . . . 23
A.3.2 Changes (non-extensive) . . . 23

B Session info **24**
B.1 Libraries used in the project . . . 24

C GNU Affero General Public License **26**
C.1 Preamble . . . 26
C.2 TERMS AND CONDITIONS . . . 27
C.2.1 0. Definitions. . . . 27
C.2.2 1. Source Code. . . . 28
C.2.3 2. Basic Permissions. . . . 28
C.2.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law. . . . 29
C.2.5 4. Conveying Verbatim Copies. . . . 29
C.2.6 5. Conveying Modified Source Versions. . . . 29
C.2.7 6. Conveying Non-Source Forms. . . . 30
C.2.8 7. Additional Terms. . . . 32
C.2.9 8. Termination. . . . 33
C.2.10 9. Acceptance Not Required for Having Copies. . . . 34
C.2.11 10. Automatic Licensing of Downstream Recipients. . . . 34
C.2.12 11. Patents. . . . 34
C.2.13 12. No Surrender of Others' Freedom. . . . 36
C.2.14 13. Remote Network Interaction; Use with the GNU General Public License. . . . 36
C.2.15 14. Revised Versions of this License. . . . 36
C.2.16 15. Disclaimer of Warranty. . . . 37
C.2.17 16. Limitation of Liability. . . . 37
C.2.18 17. Interpretation of Sections 15 and 16. . . . 37
C.3 How to Apply These Terms to Your New Programs . . . 38

Preface

The book was intended as documentation on a suggested pipeline for normalising brain and lesion masks from chronic stroke T1 weighted imaging. To make it a bid more widely useful, the pipeline documented here will register T1 weighted images to either 1 or 2 mm MNI standard space and include a lesion mask in that process, if any is present.

The pipeline was comprised as part of my exchange stay at the Brain Behaviour Lab at UBC, Vancouver, Canada during August-October 2023. It is my hope that the notes and scripts may be of help to others, myself being a new learner to FSL and everything around it.

Motivation

The reason for creating this tool is to use 1mm MNI normalised (registration) lesion masks for the NeMo tool.

Most of the hard working scripts in this pipeline are based on the work by Dr. Dianne Patterson, PhD, which in turn is also based on others work. I have tried my best at modifying the original scripts as little as possible for clarity, and instead created a few new scripts to work as wrappers.

And then, the frustration. I found that documentation that is easy to understand is highly lacking in the field (or I just didn't manage to find it). Also, the documentation on fsl-functions is difficult to always follow. As an example, this is the best overview of `fslmaths -help`. And please notice, that to get help for `fslmaths`or `flirt`, type the command followed by `-help`. If you want the same from `fnirt`, then type `-h` or `--help` as is the case for most other command-line programs.

I have tried to lower the bar to get started working on MRI registration. Feedback is very welcome on GitHub.

License

term	definition
fsleyes	Thes picture viewing tool from the fsl-package. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes
MNI	Montreal Neurological Institute. https://nist.mni.mcgill.ca/atlases/
MRI	Magnetic Resonance Imaging
NeMo	Network Modification tool. https://github.com/kjamison/nemo
pipeline	A set of tools working together as chain links being dependent on one-another. Just a fancy ter
registration	I this context, registration denotes to art of registrering to normal space. In this case to the MI

Glossary

1 Contents

The pipeline provides a set of handy tools listed below, which I will also go through:

- Bash pipeline to process multiple subject folders to do cropping, bias correction and skull stripping before registering brain and possibly lesion mask to 1 or 2 mm MNI space (this is the primary content). The registration can be performed with fsl or ANTs.
- A few other Bash-scripts to register brain and lesion mask after manual correction of brain mask (no, even using optiBET is not perfect), and to do other things.
- A few *R*-scripts to organise files and to package lesion masks to supply to the NeMo tool in Chapter 5.
- In a separate folder (`nemo/`) I've included a few scripts to collect data from the NeMo-output and do basic visualisation based on the NeMo output data.
- Finally, three Python scripts from the NeMo-repository: `nemo_save_average_glassbrain.py`, `nemo_save_average_graphbrain.py` and `nemo_save_average_matrix_figure.py`. These are used for data visualisation in the included pipeline use case example in Chapter 5. The NeMo-project project, however is shared without a license. The files can be downloaded from the NeMo-repository or directly with the following lines of code:

```
download.file("https://raw.githubusercontent.com/kjamison/nemo/master/nemo_save_avera
download.file("https://raw.githubusercontent.com/kjamison/nemo/master/nemo_save_avera
download.file("https://raw.githubusercontent.com/kjamison/nemo/master/nemo_save_avera
```

1.1 Glossary

term	definition
ANTs	Advanced Normalization Tools. https://github.com/ANTsX/ANTs
Bash	Short version: The language of the terminal console in Windows, Linux and MacOS.
fsleyes	Thes picture viewing tool from the fsl-package. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes
MNI	Montreal Neurological Institute. https://nist.mni.mcgill.ca/atlases/
MRI	
NeMo	Network Modification tool. https://github.com/kjamison/nemo
optiBET	A four step optimization of the fsl BET tool from the Monti lab at UCLA. https://montilab.ps
pipeline	A set of tools working together as chain links being dependent on one-another. Just a fancy ter
Python	A snake, or in this case refers to the free and open source programming language Python. https
R	
regex	
registration	

2 Getting started

```
source("glossary_setup.R")
```

You need to have the following programs installed:

- fsl as well as
- *R* and RStudio (or similar) installed.
- I also highly recommend the package ITK-SNAP, which can be used for some (semi-)manual mask modifications.
- You may want to use the powerful ANTs package (or the underlying `ANTsR` package) for registration. The included script, to is provided by Dr. Dianne Patterson, PhD.

2.1 Let's get to it

> **Keep a backup**
>
> Before starting, be warned. The script might override original data with modified data without warning. Please keep a backup of original files. Now you have been warned.

Script files are located in the `codes` folder like below.

```
${normalisation-pipeline}
   ...
   codes
      00norm_prep_pipeline.sh
      T1_2_MNI152_1mm.cnf
      ant_reg3_bbl.sh
      file-structure.R
      fsl_anat_alt_bbl.sh
      fsl_norm_bbl.sh
      modified_brain_mask_bbl.sh
      multi_punch.sh
```

```
   nemo-packing.R
   optiBET.sh
   prep_T1w_bbl.sh
...
```

1. Download or clone the repository to your computer and extract the folder if necessary. Move the contents of the `codes` folder to whereever you find appropriate (or just leave them for now). This is now the "codes folder" and will be referenced as `/codes/folder/` in the script examples. Edit this to run on your computer.

   Now, open a terminal window and navigate to in the parent directory of your data folder, which we will now refer to as the "root directory".

   This pipeline assumes, that your files are organised in the following way. This is not completely according to the BIDS-format, but that might be coming in the future. For now, this is the way:

   ```
   ${ROOT}
      sub01
          sub01_T1w.nii.gz (T1 weighted base image)
          sub01_lesion.nii.gz (lesion mask, optional)
      subNN
          subNN_T1w.nii.gz
          subNN_lesion.nii.gz
   ```

   If, your files are like this, just jump ahead. If your files are all on one folder, but named as above, you can use the `file-prep.R` script to organise subject files into subfolders. Open the script and edit the first three variables. Save it and then, run the following in the terminal window:

   ```
   Rscript file-prep.R
   ```

   Now the files and folders should be structured as expected.

2. *Optional*: If you want to do registration to 1mm MNI space, you need to add a new config file to fsl. If not, skip ahead. The config file is the `.cnf` file in your codes folder, and it can be copied to the correct location using the following command:

   ```
   imcp /codes/folder/T1_2_MNI152_1mm.cnf $FSLDIR/etc/flirtsch/T1_2_MNI152_1mm.cnf
   ```

   Depending on what you are doing, you'll probably be fine doing 2mm registration, but for some use cases, the 1mm registration is necessary. Here is a discussion, I found useful on doing 1 or 2 mm.

term	definition
ANTs	Advanced Normalization Tools. https://github.com/ANTsX/ANTs
BIDS	Brain Imaging Data Structure. https://bids-standard.github.io/bids-starter-kit/index.html
fsleyes	Thes picture viewing tool from the fsl-package. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes
registration	I this context, registration denotes to art of registrering to normal space. In this case to the MI

3. Now you are ready for start data processing. The script has a few assumptions. It will look for a file with the file name pattern '*[Ll]esion*.nii.gz', and assume this is a lesion mask for the T1 weighted image (which should be named '*T1w.nii.gz'). You may optionally specify the lesion mask file name pattern used, eg: `sh 00norm_prep_pipeline.sh 1mm 'lesion.nii.gz'`. If no lesion mask is in the folder, the script will just perform skull stripping and registration of the head/brain. The script will also assume you want 2mm registration and do so with fsl. To view the documentation and see a few examples run this:

   ```
   sh /codes/folder/00norm_prep_pipeline.sh -h
   ```

   Then, when ready run the main script with your desired settings. Now processing starts, and it will take some time. Sit back and relax while your computer hums away. Or do something else in the meantime. You'll get time stamps along the way to have an idea of the progress and the time needed. The output files are written along the way, so you can manually check the output while the script works. Note that your original T1 and lesion mask will have "_orig" appended as suffix and be substituted with corrected files. If the script gets interrupted, you can just restart it, and it will skip subjects already processed. Make sure you delete the output files, if you want the script to rerun on a specific subject.

4. Now it is time for quality control. Please refer to the separate Chapter 3 on this. Then, you are done and can use these normalised files however you like.

2.2 Glossary

3 Quality control

```
source("glossary_setup.R")
```

As quality control goes, this is not exhaustive, but here is my suggestion as to a minimum approach.

On paper, everything is now done. But of course, you should go through all subjects manually to check that, the masking went well, as that is the real key step in this process.

Sticking to the broader lines, the problems may be of either auto-cropping or masking. As masking goes, the script may have included too much or too little in the brain mask. I'll go through suggested solutions to all these below.

3.1 Inspection

I have come to like the ITK-SNAP tool, but fsleyes is good as well.

I would open the new "SubNN_T1w.nii.gz" in fsleyes or ITK-SNAP and overlay the "SubNN_T1w_brain_mask.nii.gz" and go through to check the masking and cropping. Check the cropping and the masking. Happy? Carry on to the next subject. Not so much? See below.

3.2 Cropping

Do not skip this step. Take a good look at the cropping, as the algorithm might have cropped out part of cerebellum (typically the issue, if any).

1. Delete all the output-files (remembering that the original files were preserved with "_orig" suffix) and renaming the original files removing the suffix.
2. Open the original T1 image and lesion mask in fsleyes, and manually crop the two with the same mask when cropping, you can save the crop mask from the T1 and load it for the lesion mask). If the two files are not cropped to the same dimensions, they won't align, and you'll be in trouble.

3. After cropping, run the `00nemo_prep_pipeline.sh` script again. It will only run in folders with the *.anat folder not present.

3.3 Masking

In ITK-SNAP, you can correct the brain mask manually. Overall, open the "SubNN_T1w.nii.gz" with `Control-G` (linux), `Control-G` (windows), `Command-G` (mac) and the "SubNN_T1w_brain_mask.nii.gz" with `Control-O` (linux), `Control-O` (windows), `Command-O` (mac). You can manually edit the brain mask by using the interpolate tool to apply changes to all layers in all three planes. Please have a look at this demonstration of the tool. Make sure that changes are made in a different label to the main label of the brain mask segmentation.

Here, I'll just go through an order of work for the two different cases:

3.3.1 Too little

This is simple. You just add the missing:

1. Using the drawing tool, include the missing parts of the brain/infarct in the new label. You can get by by just drawing on each 5-7 layers in the axial plane. Then do interpolate along the axial axis. Check that you are satisfied. If not, then `Control-Z` (linux), `Control-Z` (windows), `Command-Z` (mac). Add a few other layers of manual drawing. Interpolate again. When done, save as a new file.

3.3.2 Too much

This is also relatively simple, but has an extra step:

1. Invert the lesion mask:

```
fslmaths SubNN/SubNN_T1w_brain_mask.nii.gz -binv SubNN/SubNN_T1w_brain_mask_INV.nii.g
```

2. Follow the same steps as above to add non-brain area to exclude from the brain mask. Save the new brain mask and then invert the modified inverted brain mask again:

```
fslmaths SubNN/SubNN_T1w_brain_mask_INV.nii.gz -binv SubNN/SubNN_T1w_brain_mask_MODIF
```

3. Check that you are satisfied with the result.

3.3.3 Register to MNI space again

1. Having a new, modified brain mask, go to the terminal window again and write the following:

   ```
   sh modified_brain_mask_bbl.sh SubNN/SubNN_T1w.nii.gz SubNN/SubNN_T1w_brain_mask_MODIF
   ```

   Please notice that the naming of the modified brain mask doesn't matter.

3.4 Final quality control steps

After registering the lesion mask to standard space, please make sure, that you are satisfied with the result. You might have to correct to masking, especially if its a large lesion. In my experience, the common problem is that the algorithm have left out some of the infarcted area towards cortex. In ITK-SNAP you can add this area. If the lesion is all the way to the surface of the cortex, then make sure to overfill towards the surface (into the non-brain area). Afterwards, you can perform a three step approach to "punch" out the lesion only to the surface of the brain.

1. The following will create an inverted MNI brain mask in you source directory. Make sure that the fsl directory is correct.

   ```
   fslmaths /usr/local/fsl/data/standard/MNI152_T1_1mm_brain_mask.nii.gz -binv mni_1mm_b
   ```

2. The next is the short version of a three-step operation to 1) make sure the lesion mask is binary, 2) subtract the inverted MNI brain mask as an inverted hole-puncher, and finally 3) using the threshold function to isolate the lesion mask. Please correct the subfolder and file names.

   ```
   fslmaths subNN/subNN_T1w_MNI-1mm_lesion.nii.gz -bin -sub mni_1mm_brain_mask_inv.nii.g
   ```

3. Now please rename the old MNI lesion mask and remove the "_punch" suffix from the new lesion mask name to follow the standard naming.

If you are performing several of the "punch-out" actions, I have created a small script `other/multi-punch.sh`, that will automate this process a bit. To run it, you provide a folder name to search and a common regex pattern to search for within the folder (it includes with sub-folders). The syntax would be like this:

```
sh multi-punch.sh /sourcefolder "*_lesion.nii.gz"
```

term	definition
fsleyes	Thes picture viewing tool from the fsl-package. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes
regex	Short of Regular Expression. Provides a very powerful character search interface. There are various

3.5 Glossary

Part I

Use case #1: Finding NeMo

4 Preparations

Following the steps laid out in Chapter 2, we got all lesions normalised to standard 1mm MNI space, and they are all located in the individual subject folders, som we just need a few steps to offload work to the NeMo-tool. This was the command used to perform fsl based registration to 1 mm MNI standard space:

```
sh /codes/folder/00norm_pipeline.sh --do1mm
```

1. Now you should be ready to package for the NeMo tool. The web interface has an upload limit of 10 lesion masks. Open and edit the source-folder in the `nemo-packing.R` script. It will collect all 1mm MNI lesion masks and package in zip-files of max 10 lesion masks each and put them in the provided folder. Save and then run the following:

   ```
   Rscript nemo-packing.R
   ```

Then files were uploaded and sent to the NeMo tool server.

4.1 Glossary

```
data frame with 0 columns and 0 rows
```

5 Using the NeMo tool

This section will exists as documentation on how we have worked with the NeMo tool. Coding is very much inspired and copied from other sources. I will try to write sources and inspiration.

5.1 Package and upload

First step is to get the normalised lesion masks processed by the NeMo tool to get a "disconnectome".

The online instance of the tool offers to freely process lesion masks in batches of 10. You can use the bundled *R*-script to package lesion masks into compressed zip-batches of maximum 10.

1. Open the `nemo_packing.R` file and edit the variables in the first section, then run the script, from either RStudio or the command terminal window:

   ```
   Rscript nemo-packing.R
   ```

2. Then head over to the NeMo processing website and upload the zip-files, one at a time with your desired settings and atlas for parcellation.

5.2 Glossary

term	definition
connectome	Term used to denote the collection of connections in the brain.
NeMo	Network Modification tool. https://github.com/kjamison/nemo

6 ZIP to data set

So, here I'll show how we went from compressed zip file, from the NeMo tool, and to having a data set for further analyses.

I am fairly new to Python, but have worked in R for many years. This means, that I have written just enough in Python, and everything else in R. Small python functions will be wrapped in R scripts. Just a word of warning.

First, extract the compressed file and put it where you like. I'll denote this location as the `source.folder`.

```
# source.folder <- "/full/path/NeMo_output"
source.folder <- "/Users/au301842/NeMo_output"
```

Then we want to extract the `chacovol` data. This is extracted by unpikling `.pkl` files using a short python script wrapped in R:

```
source("nemo/unpkl-sparse.R")
unpkl_sparse(data.folder = source.folder,
             "chacovol_yeo17_mean.pkl")
```

The unpikled files are then collected and merged in a wide format.

But first, we need to handle the provided atlas `.txt` files to get ROI names:

```
atlas <- readLines("/Users/au301842/NeMo_output/Yeo2011_17Networks_NetworkNames_ColorLUT.t
rois <- do.call(c,lapply(atlas[-1],function(i){
  s1 <- strsplit(i,"_[0-9]{1,3}_")[[1]][2]
  strsplit(s1," ")[[1]][1]
  }
  ))
```

```
source("nemo/nemo-collect.R")
df <- nemo_collect(
  data.folder = source.folder,
  id.pattern = "W[0-9]{2}",
```

```
  file.pattern = "chacovol_yeo17_mean.tsv",
  roi.names = rois
)
```

I very much prefer writing function to do the data handling, but I just have to note, that these functions are very primitive due to lack of time on my end. But they are working and will provide a good foundation for further worker.

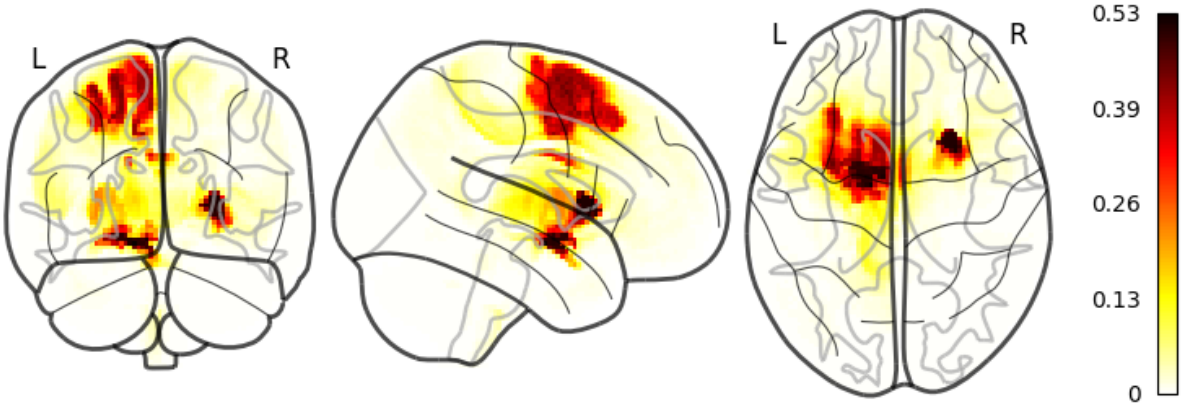
6.1 Visualisation

> ! Please note that the NeMo-repository is shared without a license
>
> This means it is technically not allowed to modify or redistribute the code. Please refer to {#contents}.

We also want those nice glass brains. These following R scripts are all simple wrappers for the scripts from the NeMo tool, which uses the `nibabel` Python package for plotting. All the examples below are created based on a small sample of two subjects and the Yeo17 atlas which is also shared in the NeMo tool source.

First we do the simpel voxel based heatmap.

```
source("nemo/glass-brain.R")
glass_brain(
  data.folder = source.folder,
  file.pattern = "chacovol_res2mm_mean.nii.gz",
  out.name = "images/glass_chacovol.png"
)
```



Next we can do the parcellation or atlas based chacovol plotting:

```
source("nemo/glass-brain.R")
glass_brain(
  data.folder = source.folder,
  file.pattern = "chacovol_yeo17_mean.pkl",
  out.name = "images/glass_chacovol_parc.png",
  parcellation = "/Users/atlas/folder/Yeo2011_17Networks_MNI152_182x218x182_LiberalMask.ni
)
```

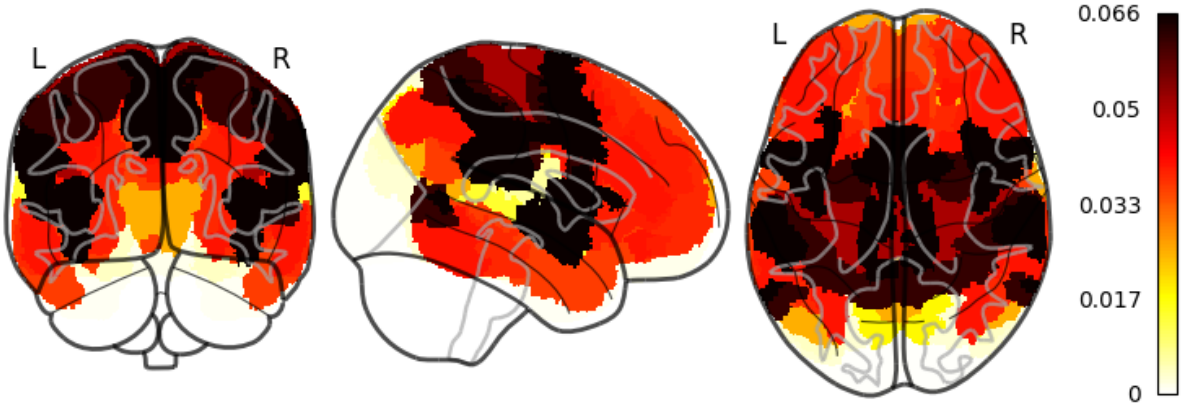


Figure 6.1: Averaged parcellation based chacovol glassbrain plot

The same scirpt can also be used for lesion plotting:

```
glass_brain(
  data.folder = "/Users/folder/with/lesions/",
  file.pattern = "lesion.nii.gz",
  id.pattern = "W(50|49)", # This id.pattern is used to only include specified ids
  out.name = "images/glass_lesion.png"
)
```

And finally the change in connections between areas in the brain can be visualised as a graph-brain:

```
source("nemo/graph-brain.R")
graph_brain(
  data.folder = source.folder,
  file.pattern = "chacoconn_yeo17_mean.pkl",
  out.name = "images/glass_chacoconn.png",
  node.file="/Users/atlas/folder/Yeo2011_17Networks_MNI152_182x218x182_LiberalMask.nii.gz"
```

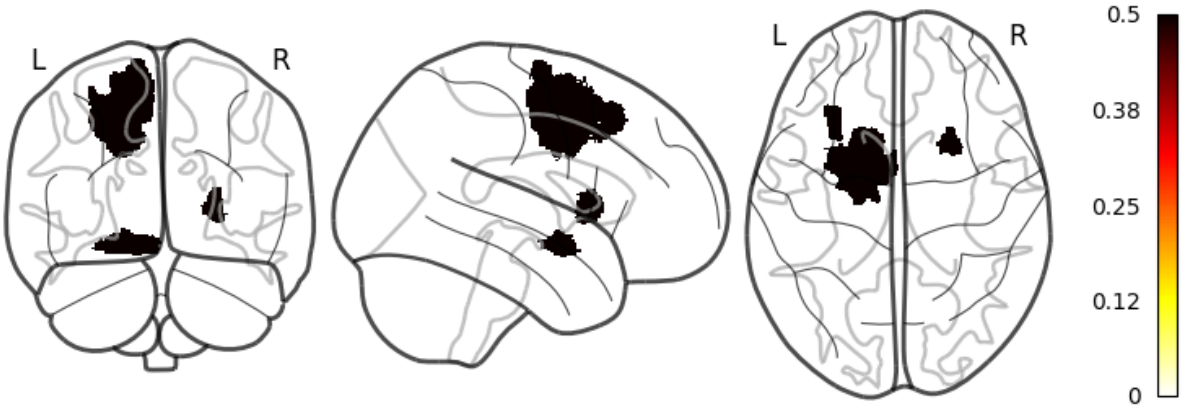


Figure 6.2: Combined lesion plot

```
)
```

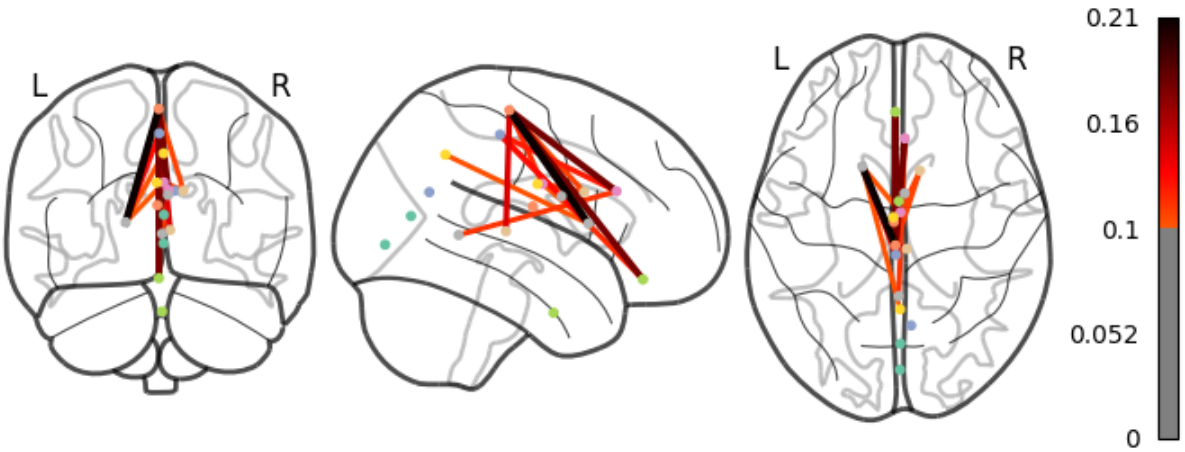


Figure 6.3: Graph brain plot of lost connections between atlas based ROIs

6.2 Glossary

```
data frame with 0 columns and 0 rows
```

A Changelog

A.1 2024.1.1

Changed versioning scheme.

As the NeMo-tool is published without a license, the package or parts of it cannot be redistributed, and should instead be downloaded directly from the NeMo repository. This pipeline-project is shared under the FOSS-license A-GPL-v3.

This version will be the first released on Zenodo and have a DOI.

DESCRIPTION file added for easier cloning of the project. Initialise `renv` with `renv::init(bare=TRUE)`.

A.2 2023.10.06

This is the final version before leaving the Brain Behavior Lab for now. I have been working hard on maturing the script to better handle different use cases.

The script will run without any specifications passed to it and default to 2 mm fsl registration with or without a lesion mask. It also 1 mm registration if you follow the instructions to supply a modified `.cnf` config file to fsl, see Chapter 2. It also does ANTs registration, but only if a lesion mask is supplied. The script also allows for lesion mask regex specification as well as regex subfolder specification. It will stopp processing if there are several lesion masks in a subfolder. The scripts in `codes/` can be located anywhere, but has to be launched from a terminal window in the parent folder of subject-wise data folder. See the Section 2.1 regarding expected file structure.

Comments and questions are welcome on GitHub Discussions.

A.3 2023.10.05

A.3.1 Notes

Restructuring of the initial book to better emphasize the more general usability of the pipeline. This will prevent renaming the repository and widen the usability. The pipeline was recently tweaked to run 1-level subfolders with or without lesion masks, and final registration space (1/2 mm) is specified on first run. The script are still a little primitive, in that they don't have a ton of control steps built in, so please stick to documentation or start modifying yourself. I'll be very happy to receive comments and PRs here on GitHub.

A few other checks has been added to make the script working a little more robustly. It will give error if several lesion masks are detected. Append data+time as suffix to original files to avoid overwriting on script re-runs.

A.3.2 Changes (non-extensive)

- 00norm_pipeline.sh: everything should now have comments. Handles missing lesion masks more elegantly. Input with named flags and help section. Runs with defaults. ANTs registration option included. Renamed.
- fsl-norm-bbl.sh: now actually does registration of lesion masks, and binarises the lesion normalised lesion mask, if it is there.
- ant_reg3.sh: script included to provide optional ANTs registration. See https://github.com/ANTsX/ANTs for installation instructions. If ANTs is installed in a virtual environment, everything should be run from within this.

B Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Sonoma 14.1.2

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  L

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Copenhagen
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.3.1    fastmap_1.1.1     cli_3.6.2         htmltools_0.5.7
 [5] tools_4.3.1       rstudioapi_0.15.0 yaml_2.3.8        rmarkdown_2.25
 [9] knitr_1.45        jsonlite_1.8.8    xfun_0.41         digest_0.6.33
[13] rlang_1.1.2       renv_1.0.3        evaluate_0.23
```

B.1 Libraries used in the project

These are the projects listed with `renv::dependencies()`

```
renv::dependencies()$Package |> unique()
```

```
Finding R package dependencies ... Done!
```

```
[1] "glossary"  "rmarkdown" "stRoke"    "zip"       "here"      "styler"
[7] "usethis"   "renv"
```

C GNU Affero General Public License

Version 3, 19 November 2007 Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

C.1 Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a

network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

C.2 TERMS AND CONDITIONS

C.2.1 0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

C.2.2 1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

C.2.3 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with

facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

C.2.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

C.2.5 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

C.2.6 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

C.2.7 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally

and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

C.2.8 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

C.2.9 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

C.2.10 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

C.2.11 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

C.2.12 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

C.2.13 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

C.2.14 13. Remote Network Interaction; Use with the GNU General Public License.

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

C.2.15 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

C.2.16 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

C.2.17 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

C.2.18 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

C.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, either version 3 of the
License, or (at your option) any later version.

This program 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.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see https://www.gnu.org/licenses/.