README skullstripper

-Installing Dependencies
-- Python
Check that python is installed on your system.
(python ships with newer Mac computers and needs to loaded on Windows)
In Command prompt (Terminal in Mac), open a window and at the prompt, type ‘python’
If it is not installed go to http://www.python.org/ (for instructions).

-- Nifty_Reg for Mac

1. Download Nifty_reg from here:
   http://sourceforge.net/projects/niftyreg/

2.Extract nifty_reg
(“unzip”—Mac-friendly unzipping software is also available from the nifty_reg download site.) In some systems, the software unpacks automatically)

In the Mac/Terminal system, we recommend placing the nifty_reg folder in the Applications folder inside the Macintosh HD, and not in the users home directory.
Instructions for creating a PATH (described in 8. below) are for this location and not
for other Applications folders in Users home directories.

3. In Terminal, navigate to the nifty_reg folder (‘cd /‘, then ‘ls’ to list Macintosh HD contents, where you should see “Applications” directory, then ‘cd Applications” and ‘ls’, which will show a list of the files and folders in this Applications directory.)

Check that “nifty_reg” appears among the list.

4. When in the Applications directory in Terminal, at the prompt, type ‘cd nifty_reg’.
This will take you to the nifty_reg directory.

5. type 'mkdir buildpath’.
This will create a “buildpath” directory within the nifty_reg directory.
This folder will receive the compiled nifty_reg programs that enable nifty_reg to run
on Linux in MAC Terminal or other operating system using Linux (nifty_reg is a C ++
program).

If this has succeeded, a buildpath folder/directory will be inside the nifty-reg folder.

6. In Terminal, inside the nifty_reg directory, type 'cd buildpath'
This will take you to the new buildpath directory.

7. type 'cmake ../'
This command will place the compiler in the buildpath directory.
You can ignore the CMake warnings, and use the default “MACOSX_RPATH” that is
enabled by default on Mac OSX.

8. type 'make'
This compiles the nifty_reg files using CMakeFiles. It will take a minute or two, and
produce a long string of Scanning, Linking and Building outputs that appear as a list
in your Terminal window.

To check that this step was successful, type ‘ls’ from within the buildpath directory,
or go to the Mac menus and click on the buildpath folder icon where you should see
a list of folders and files. Check that reg_aladin and reg_3fd appear as
“exe” (executables) inside the “reg-apps” folder.

9. Next create a path to nifty_reg/reg-apps.
After the prompt in Terminal, type:
‘echo "export PATH=/Applications/nifty_reg/buildpath/reg-apps/:$PATH" >> .bash_profile’
If this command does not give a permanent path, use this command each time you open terminal to run the program
‘export PATH=/Applications/nifty_reg/buildpath/reg-apps/:$PATH’

This command will create a path to the nifty_reg programs for the skullstripper
program to use.

To check this path, type ‘echo $PATH’ into any prompt and ensure that all folders in
the path are correctly installed and named.
Note: This command will only work if nifty_reg is in the Applications directory directly
on the Macintosh HD. If you have placed the nifty_reg program directory in another
location, you can navigate to that location and type ‘pwd’ to obtain the path.
Substitute this custom path into the export command between
“Path=“ and the “:$PATH”

9. Niftyreg should be configured to run.

-- Skullstripper Program setup
1. Download file (location tbd)
2. Unzip the file and navigate to the directory into which you unzipped ii

2. At the prompt in the directory containing the program type
   'chmod +x skullstripper.py'

-- Running skullstripper (also see Instructions in DeLora et al, Supplemental data):
We recommend creating a folder that contains a copy of the program, all the images to be stripped as .nii files, the reference images (whole head and its mask) the program will create a folder labelled "masks" to receive the new mask files and another folder "stripped_images" for the stripped images.

Note: All images must have identical headers.
The program works best when:
- the raw image is modally scaled N3 corrected using the reference image as template.
- the reference mask is refined and of high quality and from the same dataset.
- the target images are a) from the same animal at different time points; or b) from a cohort of genetically identical but different individuals; and c) captured with the same imaging parameters.
- sometimes masks are improved by performing a rigid body transformation of the raw image to the template before running the skullstripper program.

Some stripping artefacts will occur when applying a reference mask to animals that are morphologically dissimilar, as the mask alignment depends on the control point grid of the whole head image. “Fat head mice” do not align well with skinny targets. Some additional refinement may be needed to apply a reference mask to animals with different brain sizes as well. If experimental manipulation has produced large anatomical differences or intensity pattern changes, the program will also not produce the best masks.

— Commands and Usage:
Navigate to the folder containing the images and the skullstripper program.

To create a mask for a single image, type
‘python ./skullstripper.py -in <IN.nii> -ref <REF.nii> -refmask <REFMASK.nii> -out <OUTFILE.nii> -thr <threshold value>'

where “IN” is the file name of the target image; “REF” is the name of the reference head image; “REFMASK” is the name of the reference mask, and “OUTFILE.nii” is the name of the resulting mask for the target image, the -thr is a thresholding process. We recommend a value of 0.5. The -thr command will result in application of the mask to the raw target image.

To create a loop for processing multiple images, we recommend this command:
‘for f in *.nii ; do python ./skullstripper.py –in $f -ref <reference volume> -refmask <reference mask> -out Mask ${f}_mask -thr <threshold value>; done’

Glossary:
- folder and directory are used interchangeably to mean the same thing.

- ‘’ is used to indicate a command
— “” is used to quote a title of a file or folder
— <> is used to indicate generic text that should be replaced with the actual file or folder names when entering a command