CHAPTER 3: HOW TO MAKE A LINUX VIRTUAL MACHINE FOR MRI ANALYSIS

I have been through many instances of creating a stable virtual machine for MRI analysis. The best I have ever built it based on Ubuntu 16.04 LTS (Long Term Support). It is very stable and the software installs without any problems, this includes VMware Tools (so you can copy to and from the VM as though it was a directory), FSL 5.0.10, MRtrix3, FreeSurfer, and dcm2nii. However, Ubuntu 18.04 LTS is now very stable and I suggest you use this. Using these instructions:

1. Download Ubuntu 18.04 LTS iso (https://www.ubuntu.com/download/desktop).
2. Burn the iso to a DVD
3. Download and install VMware workstation player (http://www.vmware.com/products/player/playerpro-evaluation.html). Enter your email during installation if you work for a university.
4. Open the player and select “Create new virtual machine”, then point it to your DVD. Select Ubuntu, and make sure to make the virtual hard drive *at least* 60GB in size. Let the VM build.
5. Open the new VM and enter your password. First thing to do is make sure that the VMware Tools is installed. If not, select Player, then Manage, then Install VMware Tools.
6. Open a terminal, and then right-click on the new black box on the left toolbar: select the “Lock to bar” option.

Install MRtrix3:

1. Install dependencies:

sudo apt-get install git g++ python python-numpy libeigen3-dev zlib1g-dev libqt4-opengl-dev libgl1-mesa-dev libfftw3-dev libtiff5-dev

2. Clone the MRtrix3 repository:

git clone https://github.com/MRtrix3/mrtrix3.git

3. Configure the MRtrix3 install:

cd mrtrix3

./configure

4. Build the binaries (this can take a while):

./build

5. Set up MRtrix3:

From the top level MRtrix3 directory, run the following:

./set_path

6. Close the terminal and start another one to ensure the startup file is read (or just type 'bash')
7. Type **mrview** to check that everything works

When installed on Ubuntu 18.04, you may receive a message regarding a missing module called libcanberra. To fix this, type in:

sudo apt install libcanberra-gtk-module libcanberra-gtk3-module

If something goes wrong then follow instructions *verbatim*:
http://mrtrix.readthedocs.io/en/latest/installation/linux_install.html

Install FSL 5.0.11:

The easiest way to install FSL is by using the automated installer script called fslinstaller.py. It relies on Python which will have already been installed during the MRtrix3 installation. If for some reason it is not installed, then open a terminal and type in: **sudo apt install python**

You can download the fslinstaller.py from the FSL downloads url (https://fsl.fmrib.ox.ac.uk/fsldownloads_registration). After downloading, navigate to your ~/Download folder and type in: **python fslinstaller.py** . Press Enter (or sometimes Y) to accept the defaults. It's around 2GB so may take some time to complete. Some great features of 5.0.11 (beyond 5.0.9) is that it includes eddy_openmp as well as eddy_cuda. However, I suggest that you install nVidia CUDA version 8.0 and then download the eddy_cuda8.0 patch from the FSL site (https://fsl.fmrib.ox.ac.uk/fsldownloads/patches/eddy-patch-fsl-5.0.11/centos6/) and then move it to your default FSL location (e.g. usr/local/fsl/bin). Make sure to give it permission so you can execute it (e.g. **sudo chmod 775 eddy_cuda8.0**).

Before trying to use eddy_cuda8.0, first install CUDA 8.0. But, before downloading it, perform pre-installation checks to make sure your graphics card is compatible:

Verify You Have a CUDA-Capable GPU: **lspci | grep -i nvidia**

Verify You Have a Supported Version of Linux: **uname -m && cat /etc/*release**

Verify the System Has gcc Installed: **gcc --version**

Verify the System has the Correct Kernel Headers and Development Packages Installed: **uname -r**

If all reports ok, then open a terminal and type in:

sudo apt-get install cuda

Press 'Q' to bypass the first bit (or Enter to read the enter EULA), and then it will begin installing. It's about 1.2GB so may take a while. Accept the defaults.

After CUDA has finished installing, setup the environment:

export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}

To run eddy_cuda8.0, just use eddy_cuda8.0 at the beginning of the command (instead of eddy or eddy_openmp). It also has additional optional flags to allow for slice-to-volume correction and some other functionalities.

Install FreeSurfer

1. Download FreeSurfer version 6.0 from here: https://surfer.nmr.mgh.harvard.edu/fswiki/DownloadAndInstall . Select the Linux CentOS 6 x86-64 file called "freesurfer-Linux-centos6_x86_64-stable-pub-v6.0.0.tar.gz"
2. Make sure to obtain a license key (https://surfer.nmr.mgh.harvard.edu/registration.html), otherwise you won't be sent a license file to make FreeSurfer work
3. Copy the file to your desktop, then navigate to your desktop (**cd Desktop, or cd ~/Desktop**)
4. Extract and install the file:

tar -C /usr/local -xzvf freesurfer-Linux-centos6_x86_64-stable-pub-v6.0.0.tar.gz

5. Set your PATH via bashrc:

gedit ~/.bashrc

6. Go to the bottom of the file (after the FSL details) and type in:

export FREESURFER_HOME=/usr/local/freesurfer

source $FREESURFER_HOME/SetUpFreeSurfer.sh

7. Open your email and check for an email from FreeSurfer. Copy the text between the lines in the email and paste it into a text document, and save it as license.txt. Copy that license.txt file into your usr/local/freesurfer directory.
8. To test it, go to your subjects directory (usr/local/freesurfer/subjects), and type in:

tksurfer bert rh pial (the right hemisphere of "bert" should now appear).

Note: to use multiple cores/threads for recon-all, add the following to the end of your recon-all command line: -openmp number_of_cores

For example, if your CPU has 8 cores, then: **-openmp 8**

Note that not every step of the recon-all -all pipeline can take advantage of -openmp, only certain steps (e.g. mri_em_register, mris_sphere, mris_register), so using 8 cores/threads will not make the pipeline 8 times faster, but it will be faster for those steps that use openmp. Alternatively, try the **-parallel** flag instead of the -openmp flag.

Important! Make sure to give yourself permission to write to various folders, such as the /freesurfer folder: sudo chmod 775 -R usr/local/freesurfer

To be able to run the FreeSurfer **hippocampal subfield segmentation**, first install the Matlab runtime (https://surfer.nmr.mgh.harvard.edu/fswiki/MatlabRuntime):

curl (or wget)
"http://surfer.nmr.mgh.harvard.edu/fswiki/MatlabRuntime?action=AttachFile&do=get&target=runtime2012bLinux.tar.gz" -o "runtime.tar.gz"

Then unpack it:

tar xvf runtime.tar.gz

NOTE: Installing FreeSurfer on Ubuntu 18.04 can require a little extra work. If tksurfer does not work, then do the following:

Install tcsh: **sudo apt-get install tcsh**

and csh: **sudo apt-get install csh**

It is also possible that freeview will not work (type in **freeview** and see what happens). To fix this:

cd $FREESURFER_HOME/bin

sudo cp freeview.bin freeview.bin.BKP

Download the latest "development version" of Freeview using the following link. Again, overwriting the original freeview.bin file may require sudo access:

https://surfer.nmr.mgh.harvard.edu/fswiki/UpdateFreeview

Make sure the new freeview.bin file is made executable. This is achieved via the following command:

cd $FREESURFER_HOME/bin

chmod a+x freeview.bin

Then install the missing library (libjpeg.so.6):

sudo apt-get install libjpeg62 git-core

Close the terminal, open a new terminal, and type in **freeview**. It should now work.

Install dcm2nii

For dcm2nii, install MRIcron (dcm2nii is part of the MRIcron package):

http://neuro.debian.net/pkgs/mricron.html

Select **"install this package"**

Select your operating system (from the pull-down menu)

Select a download server **(any will do)**

Select desired components: **select "all software"**

2 new command lines will appear: copy and paste them into your Linux terminal

Then, type in: **sudo apt-get update**

Then, type in: **sudo apt-get install mricron&_sm_au_**

Install dos2unix

If you draft bash files (scripts) in text (.txt) format on a Windows or Mac platform, then you will need to convert them to Unix format for them to be executable in Ubuntu. So, install a small program called "dos2unix" in your VM:

sudo apt-get install tofrodos

sudo ln -s /usr/bin/fromdos /usr/bin/dos2unix

To use it, navigate to the directory where your txt file is (e.g. my_text_file.txt) and type in:

dos2unix my_text_file.txt

The file will now be in Unix format.

To make the .sh file executable, type in:

sudo chmod a+x 'name_of_file.sh'

To install 16.04 LTS, follow these instructions

1. Download Ubuntu 16.04 LTS iso (https://www.ubuntu.com/download/desktop).
2. Burn the iso to a DVD
3. Download and install VMware workstation player (http://www.vmware.com/products/player/playerpro-evaluation.html). Enter your email during installation if you work for a university.
4. Open the player and select "Create new virtual machine", then point it to your DVD. Select Ubuntu, and make sure to make the virtual hard drive *at least* 60GB in size. Let the VM build.
5. Open the new VM and enter your password. First thing to do is make sure that the VMware Tools is installed. If not, select Player, then Manage, then Install VMware Tools.
6. Open a terminal, and then right-click on the new black box on the left toolbar: select the "Lock to bar" option.

You can install FSL 5.0.11 (as described above) or *FSL 5.0.9*:

1. Go to this link http://neuro.debian.net/pkgs/fsl-complete.html. Click on "Install this package", select your operating system (Ubuntu 16.04 "Xenial Xerus"), then Select a download server. Now select "all software". Two new lines will appear (wget, and sudo): copy them into your terminal (and enter your sudo password).
2. Copy and paste the next line into your terminal ("**sudo apt-get update**")

3. Copy and paste the next line into your terminal ("**sudo apt-get install fsl-complete&_sm_au_**").
4. Set your PATH via bashrc:

gedit ~/.bashrc

5. Go to the bottom of the .bashrc file and enter the following lines:

FSLDIR=/usr/share/fsl

. ${FSLDIR}/etc/fslconf/fsl.sh *# note the space between the . and $*

PATH=${FSLDIR}/bin:${PATH}

export FSLDIR PATH

6. Save and close .bashrc
7. Type in: **fsl**

If nothing happens, then your FSLDIR is different, such as /usr/share/fsl/5.0/. So, change this line accordingly in your .bashrc file, save, close. Then close the terminal, open a new terminal, and then type in fsl in your terminal. The fsl GUI should now appear. Advanced options for installing FSL are here: https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation/Linux .

Install eddy for FSL:

1. Go to: http://neuro.debian.net/pkgs/fsl-5.0-eddy-nonfree.html
2. As for FSL, select Install this package etc and follow the instructions.
3. Now, after eddy has installed, you want to install eddy_openmp so that you can utilize more than one CPU core/thread. So, click here: http://fsl.fmrib.ox.ac.uk/fsldownloads/patches/eddy-patch-fsl-5.0.9/centos6/, and download the file called eddy_openmp. Copy it to your VM desktop, and then copy it to your fsl/bin directory using a new terminal:

cd Desktop (or cd ~/Desktop)

sudo mv eddy_openmp /usr/share/fsl/5.0/bin/ (this moves the file to the bin directory)

sudo chmod 775 /usr/share/fsl/5.0/bin/ eddy_openmp (this gives you permission to use eddy_openmp)