Instructions for installing gwemopt on UNIX

Geoffrey Mo, Carleton College
mog@carleton.edu

May 29, 2020

These are instructions for setting up the gwemopt tool primarily written by Michael Coughlin. These instructions were written for a machine running Ubuntu 16.04, but should work for most UNIX distributions. These instructions are compatible with macOS.

In these instructions, `$ ls` means to run ls.

1   Installing Anaconda

Any Python 3.7 distribution will work, but Anaconda comes with many of the packages required and can be installed completely locally, so it is convenient to use. If you prefer not to use Anaconda, skip this step, but you may have to install additional packages not mentioned in these instructions.

1. Go to `https://www.anaconda.com/download` and download the Anaconda 3.7 installer. Put it in your home directory.

2. If you are installing Anaconda on macOS, further instructions can be found in `https://docs.anaconda.com/anaconda/install/mac-os/`; otherwise, continue reading.

3. `$ cd ~`

4. `$ bash Anaconda-latest-Linux-x86_64.sh`

5. Hold down the enter key to skip through all the licensing, then type "yes" to agree to the terms.

6. We want to install this locally, so the ~/anaconda2 location given by default is good.

7. When prompted, type "yes" to prepending to PATH in ~/.bashrc. This is so when you type `$ python` in the command line you get this Anaconda distribution of Python.

2   Installing Python packages

1. `$ pip install --user lscsoft-glue healpy astroplan astropy ligo-gracedb pyephem mocpy astropy_healpix pymultinest`

3   Downloading gwemopt

1. `$ git clone https://github.com/mcoughlin/gwemopt`. I did this in my home directory.

2. To install this, `$ cd gwemopt` and then `$ python setup.py install --user`

4 Installing MultiNest and PyMultiNest

MultiNest and its Python wrapper PyMultiNest are required, but have slightly more involved installation procedures. These instructions are a more detailed version of what is written here: `https://johannesbuchner.github.io/PyMultiNest/install.html`. To begin, in your home directory, `$ git clone https://github.com/JohannesBuchner/MultiNest` and
`$ git clone https://github.com/JohannesBuchner/PyMultiNest.git`.

4.1 Installing cmake

The MultiNest installation requires a C compiler. If you have one already, feel free to skip this subsection.

1. If you are using macOS, first type into the terminal:
   `$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"`
   then run `$ brew install cmake`. Otherwise, follow the instructions below.

2. Go to `https://cmake.org/download/` and download the Unix .tar.gz tarball (cmake-3.17.3.tar.gz is the latest release at time of writing). I put it in my home directory.

3. To open up the tarball, run `$ tar -zxvf cmake-3.17.3.tar.gz`

4. `$ cd cmake-3.17-3`

5. `$ ./bootstrap && make && sudo make install`. This will take a few minutes, so go get some coffee.

4.2 Installing other dependencies for MultiNest

1. For macOS, run `$ brew install openblas`, and then run `$ brew install lapack`. Otherwise...

2. `$ sudo apt-get update` to update the list from where apt-get pulls.

3. `$ sudo apt-get install libblas-dev liblapack-dev`

4.3 Building MultiNest and PyMultiNest

1. `$ cd ~/MultiNest/build`

2. `$ cmake ..  && make`

3. To add MultiNest to your path right now and forever (by adding it to your ˜/.bashrc),
   `$ export LD_LIBRARY_PATH=~/MultiNest/lib/:$LD_LIBRARY_PATH` and
   `$ echo 'export LD_LIBRARY_PATH=~/MultiNest/lib/:$LD_LIBRARY_PATH' >>~/.bashrc`

4. `$ cd ~/PyMultiNest`

5. `$ python setup.py install --user`

6. Adding PyMultiNest to your path right now and forever,
   `$ export PATH=$PATH:$HOME/.local/bin/` and
   `$ echo 'export PATH=$PATH:$HOME/.local/bin/' >>~/.bashrc`

5   Testing the installation

Now you should have everything installed. To test this, we will run a command which will return skymaps.

1. `$ cd ∼/gwemopt/bin/`

2. In the next command, the `--doEvent` flag uses the LIGO GraceDb database and hence requires a LIGO login certificate. Do this with `$ ligo-proxy-init`.

3. Now we will run the test command.

   ```
   $ python gwemopt_run -l ../lightcurves/Me2017_H4M050V20.dat --doEvent --do3D
       --doTiles --doPlots --doSchedule  --tilesType hierarchical
       --timeallocationType powerlaw --scheduleType greedy --doEfficiency -o
       ../output/hierarchical_powerlaw_greedy --powerlaw_dist_exp -2.0 --Ninj
       1000
   ```

4. This should return plots and skymaps in the /gwemopt/hierarchical_powerlaw_greedy/ directory.