GW_Rank Documentation

Nicole Ford

May 2022

1 Introduction/Motivation

The MEGA research group uses ground-based telescopes to conduct electromagnetic (EM) followup observations of potential compact object mergers detected with gravitational wave (GW) signals. Our ability to observe a GW localization region to look for an EM counterpart depends on factors such as the capabilities of the telescope, when and were the GW event occurs, and how well-localized the GW event is.

To successfully follow up a GW event, we need to be able to quickly plan an observing strategy that optimizes our chances of seeing an EM signal from the localization region of the GW event. There has been much work to figure out optimal observing strategies, and most of this work falls into the categories of either sky tiling or ranked galaxy observations. Our research group currently uses the Canada France Hawaii Telescope (CFHT) for followup observations. We previously followed up the event GW190814 with a combined sky tiling and ranked galaxy search, but this process involved manually determining the best galaxies/tiles to observe. We'd like to automate our followup pipeline further so that future observations can be carried out more rapidly and accurately. To complete these improvements to our followup observation pipeline for use with CFHT, we present the GW Rank code (available on Github).

2 The Code

The GW Rank code has three key features: it monitors the GW public alerts network and downloads the localization skymap when a new alert is released, it then outputs an optimal sky tiling scheme and a ranked list of galaxies with the highest probability of hosting the GW signal. The tiling and galaxy ranking components of the code rely heavily on the existing *gwemopt* and *HOGWARTs* codes, adapted to work specifically with CFHT.

2.1 GW Alert

The first step of the GW Rank code is to receive a GW alert and download the relevant files for that alert. This is done through the `LIGO_obs.py` script. Public GW alerts are sent through NASA's GCN Notices. The code has an option to leave it constantly running to "listen" for new alerts, or if you already have the localization file for an alert you can feed that into the code and run it.

If in "listen" mode, the script will locate the skymap localization file and put it into a new folder called "skymaps", inside a subfolder named for the GW event. It will then use this locally stored file for all future analysis (until a newer/more localized skymap is released), so that even if you have no internet connection you can still use the file for analysis.

Once a GCN alert is received and the skymap is downloaded, the `LIGO_obs.py` scipt will execute the gaalxy ranking and tile ranking codes.

Note, the default code is set up to take a skymap you already downloaded as input; comment out the last 3 lines of code as needed if you just want to listen for GCN alerts vs testing with skymaps. You can see a list of previous signals with skymaps for the LIGO O3 run here: https://gracedb.ligo.org/superevents/public/O3/.

Also note, the code is by default configured to listen for real signals from GCN alerts - these signals are flagged as 'observations' in the alert. If you just want to test the code, or use it when LIGO is not in operation, change the 'observation' flag to 'test' - test alerts usually go out roughly every hour.

2.2 Tiling

The tiling method involves breaking up the GW event probability skymap into a series of discrete tiles to be observed. If you have a telescope with a large FOV, tiling is an efficient way to ensure observations cover the highest probability region in the skymap. The tiling portion of the code mainly uses the *gwemopt* package created by Michael Coughlin. You can find more detailed documentation on this code in their GitHub repository and their paper. Note that to use this code you will first need to install its package dependencies (listed in their documentation) and then build the package using the `setup.py` file.

When a GW alert is received and the relevant files are downloaded, both the tiling and galaxy ranking scripts are executed.

The ranked tiling process is as follows:

1. Take a `HEALPIX` skymap file and configuration file for a telescope as input, divide up the sky into a pre-defined grid of tiles based on the field of view size of the telescope.

2. Using a pre-chosen ranking method, rank the tiles based on how likely they are to contain the GW source.

3. Determine the tile observation duration and number of tiles observed (i.e. the total amount of the skymap probability region that gets observed).

4. Based on the time window(s) available for observations and other constraints provided in the configuration file, determine the order to observe the tiles. Note: there is no guarantee that you will be able to observe all or part of the skymap localization region, the tiling code just optimizes how much of the probability region is observed.

With regards to step 1, you will need to know some specifications for your telescope. Namely, the desired filter(s), limiting magnitude, exposure time(s), telescope location, FOV coverage/shape, and other features related to the movement/processing speed of the instrument. In practice, we incorporate a bit of overlap between tiles by setting the FOV to be slightly smaller than it actually is (e.g. for CFHT, we use 0.8 x 0.8 deg instead of 0.96 x 0.94 deg, excluding the "ears"). All this information should go into the telescope configuration file.

For step 2, there are several possible tile ranking methods provided with the *gwemopt* code; these methods are discussed in depth in their paper. We have chosen to use the 'ranked' method, which ranks the tiles by the integrated probability contained within them.

In step 3, for the time allocation per tile there are again several methods provided in *gwemopt*; we use the default 'powerlaw' option (though in practice we manually set the exposure time). The 'powerlaw' option optimizes the probability of detecting the GW event with N observations, which is the sum of the probability of each observation. It does this by scaling the original probability skymap based on some assumed power law relation between the GW posterior probability distribution and the predicted distance to the GW event, as well as the assumed observed luminosity of the GW event source as a function of allocated observing time. We also impose some restrictions on the time allocation based on the predicted apparent magnitude of a kilonova signal from the GW event and the exposure time we need to reach that magnitude. In practice, our exposure time for each tile is broken up into several dithered chunks in order to reduce the effects of noise/chip gaps.

Finally, in step 4 we employ a 'greedy' method for scheduling the order of observations. This method is motivated by making sure that higher ranked tiles are observed before lower ranked tiles.

2.3 Galaxy Ranking

In cases where the GW event is poorly localized (and so the probability skymap is quite large), and/or you have a small FOV telescope, a ranked galaxy approach may work better than tiling. The galaxy ranking method involves querying a list of known galaxies contained within the high probability region of the skymap The galaxy ranking portion of the code mainly uses the *HOGWARTs* package created by Lana Salmon. You can find more detailed information on the *HOGWARTs* GitHub repository and the corresponding paper. Note that the *HOGWARTs* scripts should automatically get added to your python path when `GW_GalRank.py` is executed (this happens when a qualifying alert is received).

The galaxy ranking process is as follows:

1. Take a `HEALPIX` skymap file and identify the contour containing the desired integrated probability region.

2. Download the newest GLADE+ catalog and implement a series of selection cuts.

3. Implement a cut to only store the coordinates of galaxies located within the desired probability region.

4. Calculate a final probability score for each remaining galaxy using the probability skymap, the predicted GW event distance, galaxy distance, and galaxy luminosity. Rank the galaxies for observation in order of their probability score.

5. Store the subset of sorted galaxies that are observable from the telescope's location during the allotted observing window.

As of when this code was written, the newest GLADE+ catalog referenced in step 2 was not available in any queryable online catalogs, and so it has to be manually downloaded from the website. You only need to do this once, when you first run the code. There are several lines of commented code at the beginning of the `GLADE.py` file that implement an initial series of selection cuts. After the first time running the code, just keep this cut down version of the catalog stored locally.

In step 4, the calculation of the galaxy probability score is done following the method of Arcavi et al. 2017. This method is described in more detail in the official *HOGWARTs* documentation. The basic idea is that we can assume BNS or NSBH mergers should have host galaxies, and the luminosity of a galaxy is roughly proportional to the mass, which is proportional to the number of stars in the galaxy, which is proportional to the number of mergers occurring in the galaxy. So, with these assumptions a galaxy with a brighter luminosity (specifically, in the B-band) should host more mergers and be more likely to host the GW event.

2.4 How to Run: Applied to CFHT

As applied to CFHT instruments, the tiling method is most appropriate if using MegaCam (larger FOV), and the galaxy ranking method is more appropriate for WIRCam (smaller FOV).

For running with CFHT, you can use the stored `CFHT.config` file. It contains example information on the limiting magnitude (~ 22.5, though this may need to be updated to be more accurate) at a particular exposure time ($\sim$ 1500 seconds total, would include multiple dithered chunks) in a particular filter (g-band). As mentioned earlier, we set the FOV (for CFHT's MegaCam) to be slightly smaller than it actually is. From what I can tell, regular CFHT MegaCam observations don't charge for slewing time, so we set that to 0 for our scheduling purposes. If needed you can also add a 'max_exposure' parameter (relevant if you aren't manually setting the exposure time).

Both the tiling and galaxy ranking codes are executed when you run the `LIGO_obs.py` script and an alert is received. The current built-in command for executing the tiling portion is:

```
python gwemopt_run --telescope CFHT --do3D --doTiles --doPlots --doSchedule

--timeallocationType powerlaw --scheduleType greedy

-o '../../outputs/GW_TileRank_output/"+graceid+"/powerlaw_greedy'

--gpstime "+str(float(gpstime))+" --doSkymap --tilesType ranked

--skymap '../../skymaps/"+graceid+"/"+str(file.name)+"' --Tobs "+str(T0)+","+str(T1)+"

--filters g --exposuretimes 1000.0 --doSingleExposure
```

where 'telescope' is the telescope name, 'do3D' means use the provided skymap distance estimates in addition to probabilities, 'doTiles' means generate a tiling map of the sky, 'doPlots' outputs scheduling and coverage plots, 'doSchedule' outputs an observing schedule in addition to generating tiles, 'timeallocationType' is your chosen method for allocating exposure time to a given tile, 'scheduleType' is your chosen method for picking the order tiles are observed in, 'o' is the output directory, 'gpstime' is the time the original GW event occurred, 'doSkymap' means use a downloaded skymap file (location specified by 'skymap'), 'tilesType' is your chosen method of breaking the sky into tiles, 'Tobs' is a pair of numbers representing the start and end of the observing window (expressed in terms of the amount of gps-formatted time since the event occurred; the maximum window is 24 hours, but you can do more than one window by writing multiple pairs start/end times), 'filters' is an optional specification of your filter choice (can be more than one, like 'g,i'), 'exposuretimes' is an optional specification of a fixed exposure time for all tiles (so, ignores your 'timeallocationType' choice; this can also be specified for multiple filters), and 'doSingleExposure' means don't repeat visits to the same tile once it's been observed. We currently choose to manually set the exposure time based on our own calculation estimating the peak apparent magnitude of an EM counterpart rescaled from the observations of GW170817 using the predicted GW event distance. In practice this is similar to the "When and Where (WAW)" time allocation method provided by 'gwemopt' (this method is currently broken in the code).

Note: The tiling and galaxy ranking codes are currently configured to search over the 90% probability region from the skymap (meaning, there's a 90% chance of the GW event occurring in that region).

Note: The code is currently configured to build a tiling/galaxy ranking scheme for the next available nighttime window (i.e. nighttime within the next 24 hour period).

2.5 Output Files

When the code runs for a GW alert, it outputs to two subdirectories in the 'outputs' folder; one for the tiling method, and one for the galaxies method.

The galaxy ranking code outputs a basic map of the 50% and 90% contour regions, as well as a .dat file containing the following information (in ascending column order): galaxy number in the list, galaxy name, galaxy probability score, RA (deg), Dec (deg), location probability score, distance (Mpc), distance probability score, B magnitude, B luminosity probability score, and cumulative probability score.

The tile ranking code outputs several files, but the main ones of interest are the `schedule_CFHT.dat` and the `tiles_coverage.pdf`. `schedule_CFHT.dat` contains relevant information on the schedule of pointings (in ascending column order): field identifier, RA (deg), Dec (deg), tile observing start time (MJD), limiting magnitude reached, tile exposure time (s), probability covered by the tile, airmass, and filter. `tiles_coverage.pdf` displays the map of tiles observed during the allocated observing window, and can be useful for visualizing/checking that everything looks reasonable. I also have a more advanced tile plotting script (`plot_coverage_all_separate.py`, not included in the basic code installation (the script was originally made by Nick).

2.6 Future Work

I am currently working on configuring the code so that it's also possible to do a hybrid tiling/galaxy ranking method. In particular, there may be situations where it makes sense to tile the highest probability region, and then do a targeted galaxy search in the lower probability region. For our previous observing campaign of GW190814, we had enough time per night to fully map the 50% region, but not enough to map a significant chunk of the 50% $< p <$ 90% region. Rather than tiling parts of the lower probability region without any galaxies in them, we can instead strategically choose tiles with lots of bright galaxies and in that way increase our odds of finding a counterpart.

3 Acknowledgements

I'm strongly relying on the existing 'gwemopt' and 'HOGWARTs' codes, as well as work done by Nick Vieira, John Ruan, and Daryl Haggard during the previous observing campaign.