**Post-Processing OpenQuake Output**Written by E. R. Abbott, last modified 13 June 2018

If you have not done this as part of your setup, please do the following:

1. **Adding Python Libraries to Run GNS Post-Processing Scripts**
   1. Specific, commonly used python libraries are required to run our post-processing scripts. Use the following commands to add these libraries to your engine:

sudo apt-get install python3-pyqt4 python3-scipy python3-matplotlib

Once added, you will not be required to add these packages to your engine again.

1. **Setting Your PYTHONPATH to Run GNS Post-Processing Scripts**
   1. You will then need to direct your PYTHONPATH to the appropriate location. Use the following command:

export PYTHONPATH=/usr/lib/python3/dist-packages

**WARNING:** You will need to repeat this command for every new terminal window in your engine.

In a nutshell, we do not presently have scripts to produce tables, so any data tabulation will need to be done yourself.

We do, however, have scripts that enable you to visualise various aspects of your output.

Post-processing scripts developed by GNS should be available here:

/media/sf\_OpenQuake\_Resources/Post\_Processing\_Tools/GNS/src

The corresponding folder on the I:drive (ie. accessible through windows) is:

I:\Models\OpenQuake\Post\_Processing\_Tools\GNS\src

**WARNING:** A number of scripts exist in this folder that need to be edited or are no longer in use. The following three are commonly used for visualisation at this stage. Please DO NOT attempt to use any of the others, and when in doubt refer back to these guidelines. The scripts primarily in use at this point are:

SHAP.py

MagnitudeDistanceContribution.py

NrmlOsmDisplay.py

**IMPORTANT**: if you have not done this already (which you should have) or have opened a new terminal window, you will need to have modified your PYTHONPATH in order for these scripts to run. If you get any sort of error involving PyQT4, it is likely you have not done this.

export PYTHONPATH=/usr/lib/python2.7/dist-packages

1. Hazard curves and Spectra
   1. Your output must be in xml
   2. The script that will enable you to visualise this output is called:

SHAP.py

* 1. Use the following command:

python /media/sf\_OpenQuake\_Resources/Post\_Processing\_Tools/GNS/src/SHAP.py

* 1. This should pop up a graphical user interface (GUI) that will enable you to identify the files you need and adjust parameters as required.
  2. In the GUI, select either Load Files or Load Folder. To select what you want, navigate to those files, click them so that they are highlighted, and select ‘OK’.
     1. Load Files: enables you to select specific files you want. They can come from any directory.
     2. Load Folder: enables you to select an entire directory as required.

NOTE: If you are looking at directories with a large number of files, it will take up to several minutes to load. Please be patient.

* 1. The files should appear in the top right section of your GUI. They will only appear in groups of 100 files. Hover over the filename to see the path to the file location.
  2. To sort your output so that means and quantiles are at the top of the list, click “Filename Information”, the column header. (Occasionally this closes the script, I don’t know why, we have not investigated thoroughly, but just open it up and try again).
  3. To filter the loaded files, adjust the parameters in the top-ish left, just below where the Load buttons are. (ie. you might filter to have only mean hazard curves, or to have only spectra produced for 0.198 probability).
     1. The first set of filters under ‘Restrictions’ are latitude, longitude, search radius. These are useful if there are multiple sites in a single file as it will help pare down the hazard options available. The ‘City’ selector includes major cities around New Zealand. Selecting one of these will default the latitude and longitude boxes to the coordinates of that city.
     2. The second set of filters are based off of the information available in the filename, not the contents of the file. They will allow you to filter files by the type of file you want (ie. hazard\_curve, hazard\_curve-mean, hazard\_uhs, quantile\_uhs, etc.)

NOTE: InputType options for hazard\_map are available here, but the program cannot do anything with them.

* 1. To select the files you want, click the +/- button (or something like that), next to the file on the top right. To select them all, click the ‘Selected’ label at the top of the column. Selected files will appear in the bottom right section.
  2. Using the spin boxes in the bottom right section, adjust the line type (ie. solid, dot dash), colour, line thickness, and (by ticking the box) if the data points that exist appear in larger, emphasized dots.   
     You can enter the Legend text for whichever files you like on the far right. Doing this will tick the Legend box, and the description(s) typed will be part of the legend when you graph the data.   
     Selecting the Legend box alone will create a default label (ie. one based off the filename) for selected files.   
     If you would like a default label for them all. Click the ‘Legend’ column label, which will select that option for all files. Similarly, clicking the label again will unselect all files from the Legend, even if you have typed in your own text label.
  3. The axes ranges will automatically adjust to the data in the selected files. Please feel free to change this (ie. for the y-axis, you may want it to start nearer to zero, at 0.01 or something; for the x-axis, you may want to have the axis go from 0.01 to 4.0s). Axes labels are not adjustable and will be based on the data. The program knows if it’s looking at a hazard curve or uniform hazard spectra and will adjust accordingly.
  4. Ticking the Show box under Grid means that you will have major and minor gridlines. This should be defaulted to ticked.
  5. The Smooth selection is NOT the smoothing process we do in fortran. It is a computed algorithm. I usually just leave the default (which I think is not selected)
  6. The LogX, LogY enable you to change how you are viewing the axes (ie. in log form or not).
  7. Since PGA is defaulted to 0 by the program, we have included the ability for you to choose what that lowest value is using the X Zero an Y Zero boxes. Our default for X is 0.03, which is likely most important. These are only important if the LogX and/or LogY boxes are selected.
  8. Graph. (Again, please be patient. This might take several minutes depending on the number of files you are attempting to plot).
  9. Adjust the size of your plot. We’ve not implemented anything to get uniform size. I usually place the plot window in the bottom right and then drag it to the far side of the O in the OQ label (the wallpaper of your virtual machine) and about 3.5cm above the big white box.
  10. Save the image with a name of your choice (preferably a reasonably descriptive one) in your working directory. You will have to navigate there once you select Save As.

1. Magnitude-Distance sets from Deaggregation (Disaggregation)
   1. Your output must be in csv.
   2. You will most likely want to have run the disagg post-processing script obtained from Lizzie (as it’s not currently product-ized and still has some bugs) to get a percent contribution file.
   3. The script that will enable you to visualise this output is called:

MagnitudeDistanceContribution.py

* 1. Use the following command:

python /media/sf\_OpenQuake\_Resources/Post\_Processing\_Tools/GNS/src/MagnitudeDistanceContribution.py

* 1. This should pop up a graphical user interface (GUI) that will enable you to identify the files you need and adjust parameters as required.
  2. Load the file you need from your deag output folder. You will need the csv file that has both Mag and Dist in the filename (only those, not trt, not eps, not lat, not lon).
  3. You can give it whatever title you want. There should be a box somewhere. You can label the axes as you please as well, but be aware that the default z-axis is “Contribution (%)” which may not be appropriate for all input files.
  4. A plot should appear (can’t remember if you press a button or it happens automatically…)… You can actually rotate this to whatever angle you need, which is cool. Everything should be automatically labelled.
  5. You should be able to close the plot and adjust the extent of the axes if you have a lot of squares with no contribution. Replot once you’ve done this.
  6. Expand your plot from the bottom left corner through to the O in OQ and about 3.5cm above the white box and Save. You will need to navigate to where you want to save the plot.

1. Site to Source Distance
   1. The script that will enable you to obtain this output is:

SiteSourceDistance.py

* 1. It will pop up a GUI. You will need to select a source model file to use.
  2. Use the latitude and longitude boxes to fill out your location of interest and the radius within which sources will be included.
  3. Do the analysis by clicking “go” or whatever it is. Watch the bottom bar for text telling you the analysis is complete.
  4. Use the bottom file bar to save the analysis in a csv file. Clicking on the box will pop up a window that will allow you to name and save a file for this purpose. Once you have done this, select “ok” or whatever it is to close the box with the file name and path.
  5. Click ‘Write’ to actually write the data to the file you specified.

1. Contoured Hazard Map

While this is reasonably fully fledged, I would take it with a grain of salt, not because I reckon it’s doing things wrong, but because I reckon we could make it nicer (prettier), etc. in the long term and it hasn’t been used often in the last year since its development.

This program sets the stage really well for something we are looking to do, but at this stage I would probably recommend its use for testing/visualisation to get an idea of what your hazard looks like. I think Nick H has some scripts that produce nicer looking plots with colours and such, but not 100% sure on that. That being said, it does work, so use it for whatever you want!

* 1. Your output must be in xml or csv
  2. The script that will enable you to visualise this output is called:

NrmlOsmDisplay.py

* 1. Use the following command:

python /media/sf\_OpenQuake\_Resources/Post\_Processing\_Tools/GNS/src/NrmlOsmDisplay.py

* 1. This should pop up a graphical user interface (GUI) that will enable you to identify the files you need and adjust parameters as required.
  2. Load your preferred hazard map file (in xml).
  3. We’ve only been able to get the number of contours to adjust so far, rather than the specific size of contours (ie. 0.5 step contours).
  4. When plotting, the program may take a moment or two to load. It uses the data in the file you’ve selected to go out to Open Street Map (an open source google map -esque program) to get the base layer in use.
  5. Expand to whatever size you like and be sure to save in the appropriate directory with a descriptive name.

1. Sorting hazard curves versus a fragility curve
   1. SHV3.py. Hazard curve and fragility input must be in xml.