# Geological Computing
# Making Maps using Generic Mapping Tools (GMT)

Connor

February 13, 2017

## What is GMT?

GMT stands for Generic Mapping Tools. GMT is a suite of programs developed by two graduate students at Columbia University, Paul Wessel and Walter Smith. Although they are long done with graduate school, their program is following them around. *Lesson 1* - never write a computer program that other people will find useful! You will spend a great deal of time improving it for other people. *Lesson 2* - if someone writes a useful program - USE IT! GMT continues to be an incredibly useful mapping tool for geologists and geophysicists. It is used to create location maps, plot data in numerous formats (*xy* plots, contour maps) and to do simple data analysis. Some of the geophysics tools in GMT are quite sophisticated. You can think of GMT as a geographic information system - like Grass, ArcView and similar products - only tailored to suit the needs of people who plot and present geological and geophysical data on maps.

An important difference between GMT and GIS software is that the output from GMT is a postscript file. Postscript is a language that is used and interpreted by other programs (like Ghostview, Evince, and Adobe Acrobat Reader and Illustrator) to plot images on the computer screen and to print them. Unlike GIS software, you cannot query maps once the postscript is created. Occasionally this is a disadvantage, but for the great majority of applications in geology, this is not a serious drawback. GMT produces great figures for journal articles. GMT is freely available to use and abuse. You can download it onto any computer you wish.

## Why Use GMT?

We will use GMT in this class to plot and view various types of data, including contour plots, on graphs and maps. Figures 1 and 2 were both created using GMT.

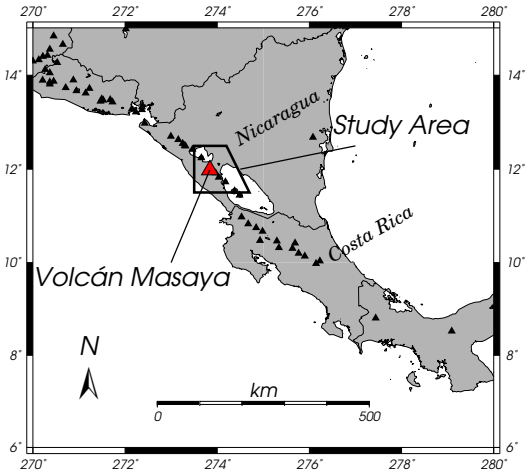


Figure 1: A simple location map generated using GMT commands. The triangles show the locations of historically active volcanoes in Central America.

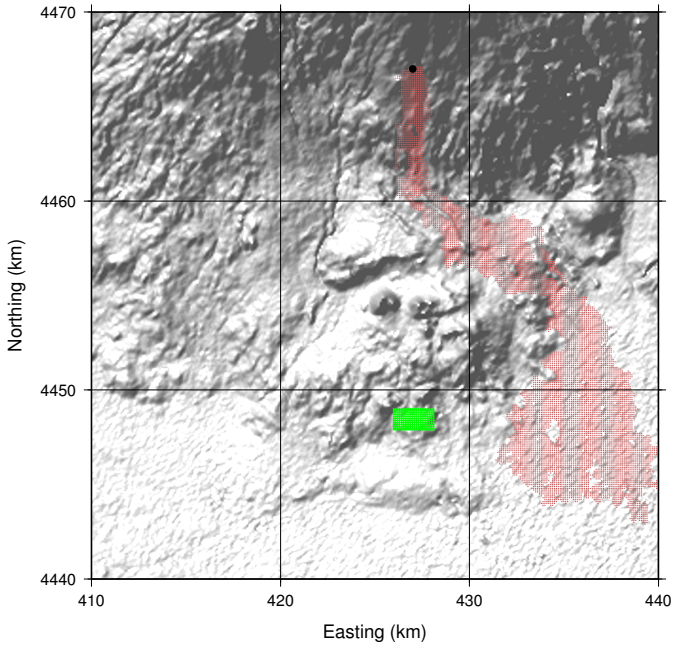


Figure 2: Lava flow (red) plotted on topography. Black dot represents active vent. Green square is nuclear power plant.

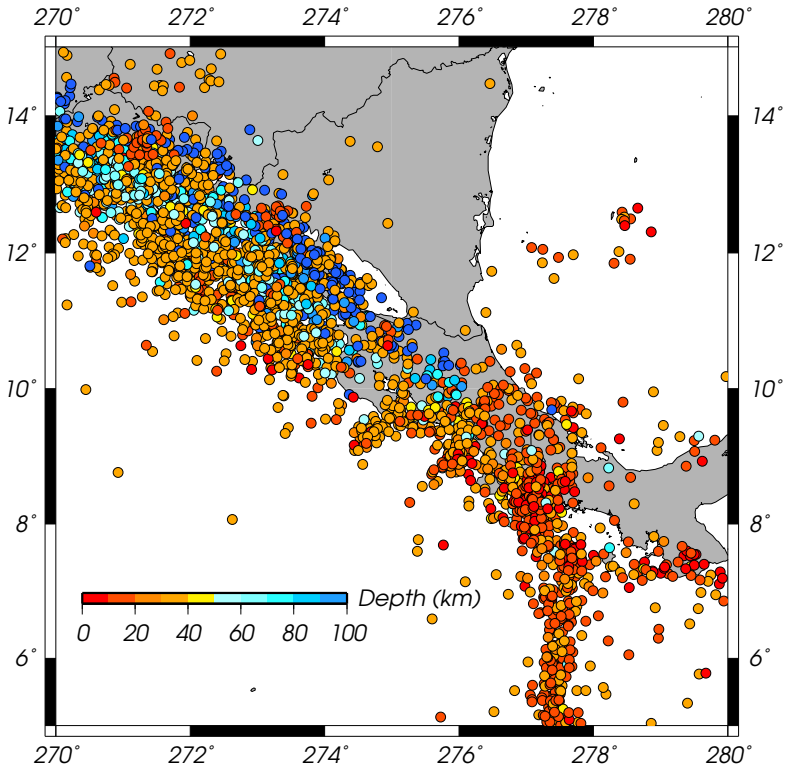


Figure 3: Earthquake hypocenters in Central America

# Simple example using GMT

The simplest way to use GMT is to just type the GMT command directly on the command line. If typed on the command line:

```
gmt pscoast -JM5i -R-88/-78/20/30 -Df -N1 -W1p -G200 -Ba2 -BNSEW -V > map.eps
```

the GMT program *pscoast* will execute and output will be directed to a postscript (EPS) file *map.eps*. Try it! You can quickly and easily view the image file *map.eps* by clicking on it. Most linux systems have image viewers that will display a postscript image. Often it is convenient to convert the postscript image to a PNG image. PNG images are smaller in size, easy to include in reports, and easy to email. To convert your map.eps image file to a PNG file (map.png) type:

```
gmt psconvert map.eps -A -P -Tg -V
```

You will notice a new file in your directory, *map.png*. Click on it to view the new image. Notice that the file size of the PNG image is smaller when compared to the size of the EPS image. This is because all editing capablility has been deleted from the PNG image.

The objects in the EPS image can be edited or changed by programs such as *Adobe Illustrator* that open and allow editing of postscript files. Using this technique it is possible to label the x-axis and y-axis, to label features on your map, to adjust line thicknesses, to overlay additional map data, etc. The possibilities are only limited by your imagination. Remember, a picture is worth a thousand words, *but only if you design it well.*

Another GMT program that is useful when making maps is *psxy*. This program plots points or locations on a map. Let us add a useful location to our *map.eps* file.

First create a text file, *location.xy*, with the following two numbers typed on one line:

```
-82.458611 27.947222
```

As you have probably guessed, these two numbers represent a location in units of degrees. Now, we need to execute our previous command again, with a slight change to the command line:

```
gmt pscoast -JM5i -R-88/-78/20/30 -Df -N1 -W1p -G200 -Ba2 -BNSEW -V -K> map.eps
```

Now, execute the following line:

```
gmt psxy location.xy -J -R -St0.4c -Gred -O -V>> map.eps
```

Another command useful for labeling locations or points drawn on the map, is *pstext*. First we need to modify the location.xy file. The one line should now read:

```
-82.458611  27.947222  Tampa
```

Type all three GMT command again. The *pscoast* command line stays the same, but note the change to the *psxy* command line.

```
gmt pscoast -JM5i -R-88/-78/20/30 -Df -N1 -W1p -G200 -Ba2 -BNSEW -V -K> map.eps
gmt psxy location.xy -J -R -St0.4c -Gred -V -O -K >> map.eps
gmt pstext location.xy -J -R -F+f9p,1,0+a0,+jLT -V -O >> map.eps
```

A critical step in using GMT effectively is to understand the parameters that are passed to each GMT command. Some parameters are required, others are optional. Notice that the **-J** and **-R** parameters are used by each of the GMT commands. The **-J** parameter specifies the map projection and the map scale. In this example the parameter **-JM5i** directs *pscoast* to draw the map using an equal area **M**ercator projection with a map size of **5 i**nches. The **-R** parameter specifies the region of interest or map boundaries. The order of the boundaries is always: West/East/South/North. These are the most important GMT parameters and used by *almost* all GMT commands.

Two other important parameters are the **-K** and the **-O**. Notice that these parameters must be included when your map involves more than one GMT command (in this example 3 commands were used to make the map). The **-K** parameter *must* be included with each command *except* the last one; this parameter is a flag that indicates the map in not yet finished and to expect more commands. The **-O** parameter *must* be included with each command *after* the first command. This parameter indicated that the command is an overlay of information onto an existing map. These parameters control the operation and synchronization of multiple commands. So if you can not view your map always check that the **-K** and the **-O** parameters are where they should be.

Notice the $>$ and the $>>$ symbols. From previous assignments you know that the $>$ symbol redirects output to a file. In this case the output file is a postscript image. Try running the *gmt pscoast* command without the redirection symbol. You will see lines and lines of postscript code. This postscript code is interpreted by postscript viewers and certain printers to produce images. The $>>$ symbol is similar to the $>$ symbol; the $>>$ sybbol *appends* information to a file. So, for a series a GMT commands, the first command *creates* a postscript image file and each subsequent command *adds* additional information to this postscript image file.

Guess what will happen if you use the $>$ symbol instead of the $>>$ symbol. Try it! This is a very common mistake, so be ready to check for it if your output is not what you expect.

The best place to get information about GMT is from the GMT website, `http://gmt.soest.hawaii.edu/`. Here you can find an FAQ and documentation about all of the commands and all of the parameters. Each GMT command has a manual page that can be accessed online or from your own command line. To access the command line manual page for *pscoast* just type *man pscoast*. The online, html version is more readable than the command line version but they both contain the same information. The online GMT documentation can be found at `http://gmt.soest.hawaii.edu/doc/latest/index.html` Here you can also find a tutorial, a listing of all of the manual pages, and a technical reference and cookbook, *very useful information for homework assignment 6*. So, bookmark the manual pages for *pscoast*, *psxy*, and *pstext*. These html manual pages are often installed to the local computer as part of the GMT package. Maybe you can find them, but, if not, they are always available online.