Git, Gnuplot and Vim Tutorial

Oliver Thomas

Quantum Engineering CDT
University of Bristol

September 2, 2018

Overview

- Version control using Git
- Plotting using Gnuplot
- Text editing using Vim

Why you should use version control

- Does this seem familiar?

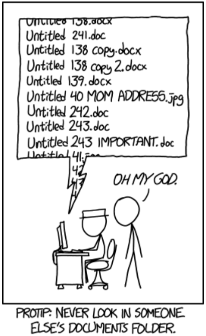

Figure: Bad version control[1]

[1] https://xkcd.com/1459/

What is Git?

- Git is one of most used version control software in the world
- Git is cross-platform and easy to use [2]

[2] https://try.github.io/levels/1/challenges/1

What is GitHub?

- Github is a cloud service for git which lets you store your repository online
- Why would you store your repository online?
  - Working remotely
  - Collaborative work
  - Hard drive failure!

Making a repository

- You can do this online on the Github website [3]
- Create a new repository
- Then click clone to get the url, open git on your computer and type:
  `git clone url`

[3] https://github.com/

Making a repository

- Go to the folder and right click `git with bash`
- You are now able to use bash for the rest of the talk!

Basic Git commands

- There are four[4] important commands you will need for git:
- `git pull`
- `git add *`
- `git commit -a`
- `git push`

[4] I cheat here and write a bash script which does these in order so I only have to run a single command.

Adding your first commit

Every repository should contain a readme, make one now then run:

- `git add *`
- `git commit -a`
- `git push`

Or use the windows GUI version and commit them to your repository.

Branching

- Branching is useful, it lets you test something out separately to the main branch.
- To make a new branch called `test`

  ```
  git branch test
  ```
- You can check all of the current branches and which branch you are on with

  ```
  git branch
  ```

Branching

- To switch to the test branch type:

```
git checkout test
```

Adding Collaborators

- Go to a repository and on the settings tab click collaborators, you can then search using a github username

Advanced Git commands

- One of the great things about Git is that you can get by with just the four (main) commands mentioned earlier.
- The git man page is very useful, especially,

  ```
  man gittutorial
  man giteveryday
  ```
- `giteveryday` is a super useful collection of the 20 commands you will need regularly.

Gnuplot

- Gnuplot is popular, multi-platform and standard software on computing clusters[5]
- https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.4

Now you will need to clone my repository,

```
git clone https://github.com/OFThomas/cdt_tutorial
```

[5] standard on most of the popular linux distributions

Example 0 Quick plotting

- Go to the `src` folder
- open gnuplot and type

```
plot 'data.txt'
```

Example 0 Quick plotting

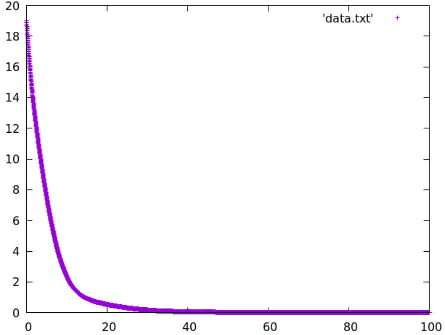

Figure: Plotting data

Lets you very quickly see what the data is doing

Example 1 Plotting functions

- Go to the `src` folder
- open gnuplot and type
  `load 'ex1_basic.p'`

Example 2 Saving plots

- Go to the `src` folder
- open gnuplot and type
  `load 'ex2_saving.p'`

Example 2 Plotting functions

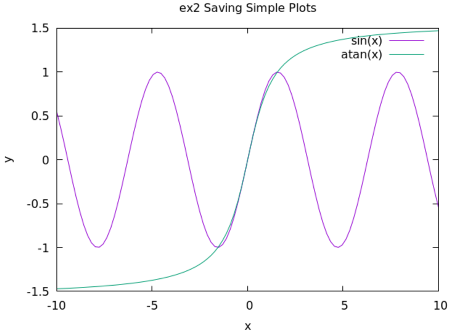

Figure: Plotting functions

Produces a png

Example 3 Bar charts

- open gnuplot and type `load 'ex3_barchart.p`

Example 3 Bar charts

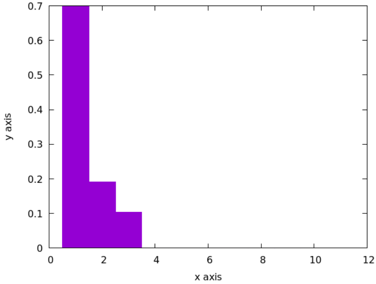

Figure: Plotting a bar chart

Example 4 Subplots

- open gnuplot

  `load 'ex4_multiplot.p'`

Example 4 Subplots

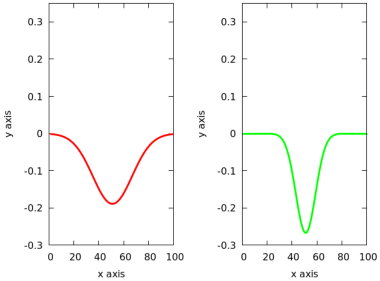

Figure: Plotting multiple subplots

Example 5 Surface plots

- open gnuplot

  ```
  load 'ex5_splot.p
  ```

Example 5 Surface plots

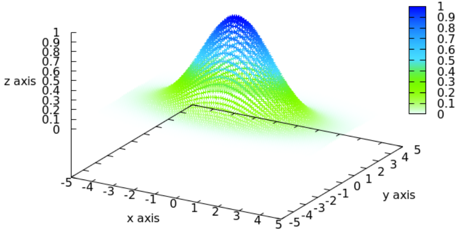

Figure: Plotting 3D data

Gnuplot summary & features

- The documentation is very good, there will be an example of whatever you want to do somewhere
- You can set *pointstyle*, *linestyle*, and colours
- Very easy to generate quick plots
- Scripts makes it easy to generate nice figures
- You can make GIFs

A brief note on text editors: Vim

- Vim is a powerful cross-platform text editor, released in 1991 and is still regarded as one of the most popular editors [6].
- Flexible with thousands of plugins available e.g. I use Vim to compile latex documents, this presentation was written in Vim.
- Computing clusters normally only have CLI so if you are running high performance code you will need to be familiar with Vim, Emacs or Nano.
- Overleaf supports Vim keybindings
- You can feel like a Hacker.

[6] Along with Emacs

Vim commands

- The most important thing to remember is that Vim has two main modes, *NORMAL*, `ESC` and *INSERT*, `i`
- All commands are run from *NORMAL* mode using `:`
- to quit use, `ESC:q` (meaning go to *NORMAL* mode, `:` means command and `q` is quit without saving)
- to save and quit use, `ESC:wq` (`w` stands for write)

In case everything goes wrong, `:q!` is force-quit without saving

Vim commands continued

All of these commands are case-sensitive and must be run in *NORMAL* mode not *INSERT*
`v` puts you in visual mode, useful for highlighting a block of text to copy or cut and paste

- `y` -yank (copy), `yy` -yank (copy) whole line
- `d` -delete (cut), `dd` -delete (cut) whole line
- `p` -paste after cursor, `P` -paste before cursor
- `x` -delete character
- `u` -undo
- `CTRL R` -redo

Movement commands

- `a` -append at the end of the next word, `A` -append at the end of the line
- `o` -open line bellow in *INSERT* mode, `O` -opens line above in *INSERT* mode
- 0 -go to start of line
- $ -go to end of line
- { -go to previous paragraph
- } -go to next paragraph

Searching and editing in Vim

Searching

- `fx` -find next occurrence of x in text,
  e.g. `f`*b* finds the next letter *b* in a line
- /x -search the whole document for x,
  e.g. /*b* finds all letter *b*s

Use `n` to go to next occurrence, `N` to go to previous.

Editing

- `r` -replace character,
  e.g. `r`*a* replaces character with *a*
- `~` -Changes the CASE of character,
  e.g. `~` when the cursor is over *a* will change it to a capital, *A*
  can be used with *VISUAL* mode to block capitalise or block lower-case

Using SED in Vim

Sed stands for Stream EDitor and can be used directly from vim [7]

Probably the most regularly used sed commands you will need are,

- :%s/foo/bar/g -replaces all instances of foo with bar globally
- :4,31s/foo/bar/g -replace instances of foo with bar in lines 4-31

[7] http://vim.wikia.com/wiki/Search_and_replace

Vim summary

- You should try Vim, it is available in overleaf
- It takes some getting used to, but I and many others think it is worth it.
- Cross-platform and powerful
- Very good documentation

Conclusion

- You should use version control
- I recommend gnuplot as it is easy to use
- Vim is a fantastic editor but does require a small amount of effort to learn

Thanks for listening!

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Figure: If it all goes wrong . . .[8]

[1] https://xkcd.com/1597/