

Weekly Report

June 4 - June 12

Christina



GitHub Tutorials





“Hello World” Project

1. Basics:
 - a. Create a repository
 - b. Create a branch
 - c. Make and commit changes
 - d. Open a pull request
 - e. Merge pull request
2. Mastering markdown & documenting projects
 - a. README
 - b. wikis
3. Creating a GitHub page
 - a. <https://chchen123.github.io/hello-world/>
4. Assigning a DOI to my “Hello World”
5. Basic Git commands
6. Issues



Linux Command Line




<https://ryantutorials.net/linuxtutorial>

1. Locate / change directories (`pwd`, `ls`, `cd`)
2. Manual pages (`man`)
3. Creating/removing/copying/moving a file/directory (`mkdir`, `rmdir`, `touch`, `cp`, `rm`)
4. Vi text editor - can be used to edit files (`vi`)
5. Wildcards - useful for searching files (`*`, `?`, `[]`)
6. Permissions (`u g o a + - r w x`)
7. Filters (`head`, `tail`, `sort`, `nl`, `wc`, `cut`,...)
8. Grep and regular expressions (`egrep`)
9. Piping and redirection (`>`, `>>`, `|`, `2>`,...)
10. Bash scripting (`echo`, `#!`, `./`, `#`, `$`,...)
11. Install atom

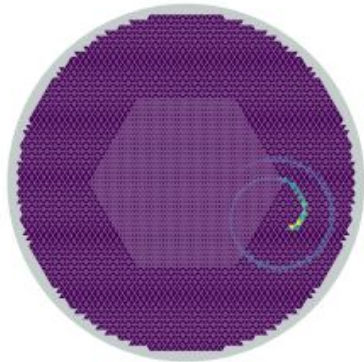


Event Simulation



- 
- Copied relevant data files to /home/chen
 - Recreated Jack's SimTest_p.ipynb work, added detailed comments for the codes
 - Renamed file as Event Simulation and uploaded file onto Data-Simulation-and-Visualization repository on GitHub

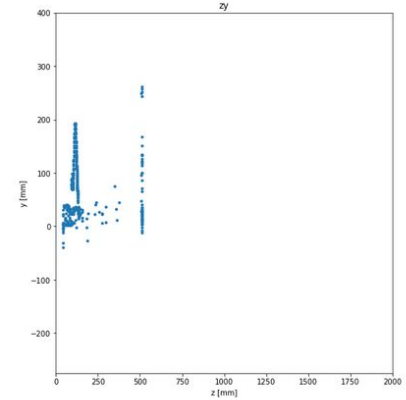
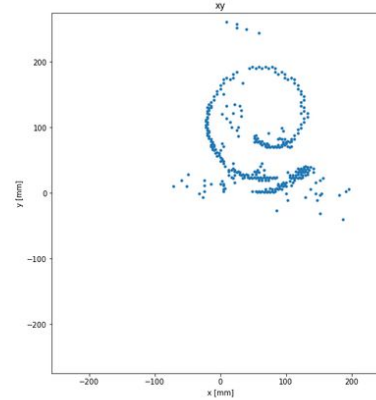
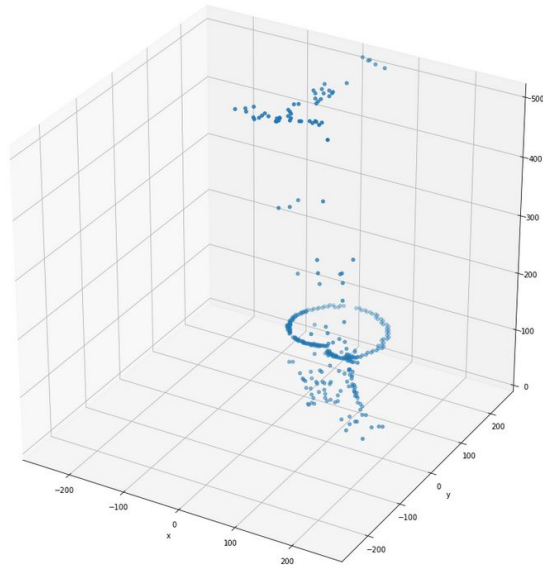
<https://github.com/chchen123/Data-Simulation-and-Visualization/blob/master/Event%20Simulation.ipynb>



Real Data



- Copied relevant data files to /home/chen
- Recreated Jack's HDF5Compare.ipynb (and RealDataLabeling.ipynb for some plotting codes) work, added comments for the codes
- Renamed as Run_0130.ipynb and uploaded to GitHub repository
- Created similar analysis for Run_0085.h5 and Run_0102.h5
- Question: couldn't find the **Peaks** and **Less** data for those runs - what do they mean?



e.g. Run_0085
Event 228

Monte Carlo Method





- We need to test quantitatively how well the modeled track fits the data
- Main class: MCFitter -> return the results ('x0', 'y0', 'z0', 'enu0', 'azi0', 'pol0') as best fit parameters
- Starts with a seed point
- Generates parameter sets (candidate tracks) from a uniform distribution over x0, y0, z0, enu0, azi0, pol0
- Each of these tracks is simulated & the one with the smallest chi-square value is selected
- Parameter space is re-centered around the selected track parameters
- The parameter space gets compressed in each dimension using the reduction factor
- After a certain number of iterations the best track is accepted as the fit result



Unfinished work

- Organize files in /home/chen and change paths for GitHub codes using them
- Make folders in repository and document in README