1. Versions

V1 2014-10-27 MS

2. Background

The IPython Notebook is a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document. This as advantageous for scientific computing applications as it allows a detailed documentation of both an experiment and analysis in one document. ipython notebook can be run either locally on your computer if you have downloaded the software below or by accessing the lab computer and using it as a server for you data.

3. Installation and required files

Required software (all open source and free):

- python, ipython, numpy, scipy packages. This is easily downloadable in the Anaconda python distribution (Continuum Analytics, http://continuum.io)
- Internet browser (preferably chrome available at http://www.google.ca/chrome/)
- pandoc (to make PDF records, http://johnmacfarlane.net/pandoc/), for Linux users this should already there, if not type `sudo apt-get install haskell-platform` in the command line.
- MiKTeX for windows (http://miktex.org/), BasicTeX for Mac (http://www.tug.org/mactex/morepackages.html),
- SSH for windows, TeX Live for Linux (http://www.tug.org/texlive/ or `sudo apt-get install texlive` in the command line).
- EyeFrictionAnalysis1-3.ipynb located at (TARDIS2\Dept1\Tannin Schmidt Group\Mike\ipython, or online at EyeFrictionAnalysis1-3). NOTE: this is version 1.3 created on October 25, 2014 please update version and SOP if changes are made.
- .xlsx summary file dasfas.xlsx located at TARDIS2\Dept1\Tannin Schmidt Group\Mike\ipython

4. Performing analysis

4.1 Using the lab computer as a notebook server

The lab computer is set up as a server for all ipython notebook. This is useful if your own computer lacks power for complex analysis or if you haven't installed the above software, you simply need an internet connection and browser.

….

4.2 Using the windows or linux/mac command shell

The easiest way to complete this analysis is to navigate and initialize the notebook files in the command shell.

1. To open a command shell in windows (if there is no shortcut) simply type cmd into the search bar on the start menu.
2. This will open the shell in your \C: directory
3. To view a listing of all the folders in your current directory type `dir` into the command line and `<enter>` to execute any commands.
4. To change directories type `cd` and then the directory of choice e.g. `cd Documents`
5. To go back a level type `cd ..`
6. If you want to work off of the network on the lab computer, first make sure you are in the home directory by typing `cd \` then change to the \Y: or other directory were the network is mapped.
7. Once you are in the directory containing the desired .ipynb file type `ipython notebook --ip=localhost` to start the ipython notebook using your own computer as the host. This will start the notebook in your browser.
8. To generate PDF document records of your notebooks type `ipython nbconvert --to latex --SphinxTransformer.author='Your name' --post PDF file.ipynb` in the command line. if you type `*.ipynb` this will convert all .ipynb files in the current directory. .tex and .pdf files are generated. The SphinxTransformer.author='Author' command sets the name of the author in the output.

4.3 Using the and editing the notebook.

4.3.1 Markdown cells and HTML basics

The primary HTML commands you will need to edit this document are

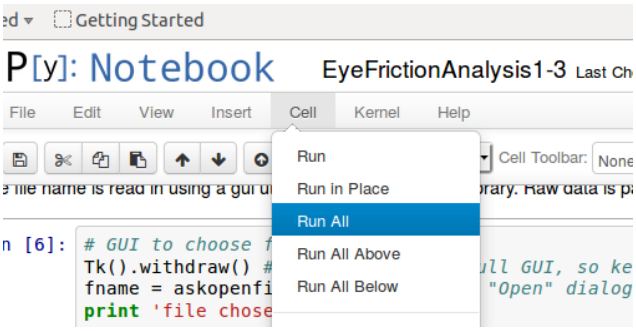
- headers `<h1></h1>` where the number denotes the level of the header. The slash command denotes the end of a formatting block.
- unordered list `<ul></ul>` creates a point form list.
- list item: `<li></li>` populates a list.
- a full html element reference is located here https://developer.mozilla.org/en-US/docs/Web/HTML/Element.

Specific documentation for your experiment and analysis is built into each notebook, in markdown cells. In the notebook there will be a General documentation cell that must be filled out and updated for each experiment. Many of the cells can be copied within repeats of the same experiment please familiarize yourself with the contents of this cell. Prior to the data analysis cell there is a full description of the analysis done. Equations are in latex math format http://en.wikibooks.org/wiki/LaTeX/Mathematics and must be enclosed on both sides by $$. If the analysis is updated please update this documentation. NOTE inline equations by wrapping the equation with a single $ will not render in PDF.

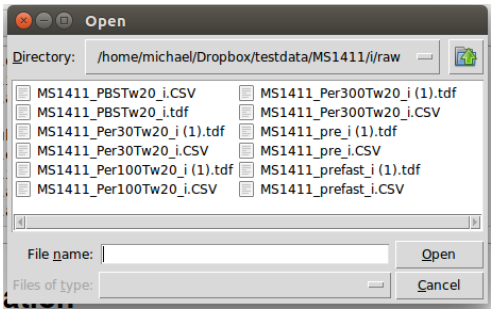
4.3.2 Using the ipython notebook

Once documentation has been updated one simply runs the notebook for a certain data set.

1. Run the notebook by selecting `Cell --> Run All`.

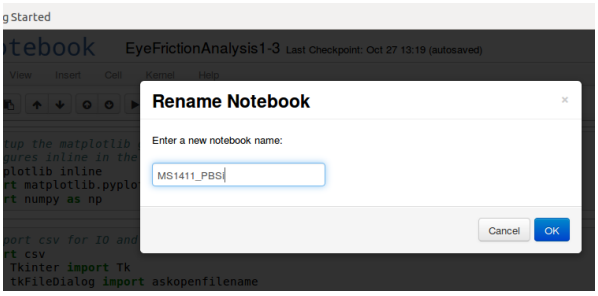


2. Choose .csv output file in the pop up window. (This window spawns behind the browser in windows)

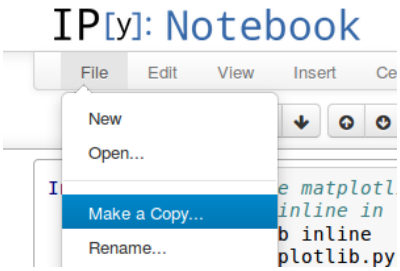


3. Analysis will run and a csv file called \FrictionAnalysis_output.csv will appear in your current working directory. Rename this file to correspond to your analysis file.

4. Renaming the notebook: click name at top of the file next to the IP[y]: Notebook icon



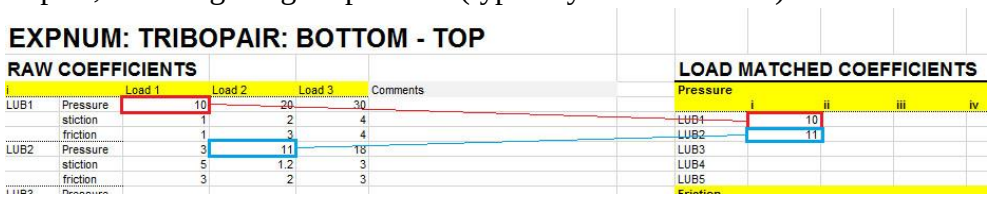
5. Making a copy of the notebook: `File --> Make a Copy`



6. Close the ipython notebook by closing all notebook tabs and typing `ctrl-c` then `y` in the command line.

4.4 Populating the excel summary

1. Open excel summary files and all .csv files outputted by the ipython notebook.
2. Update the lubricants in repeat i in the **RAW COEFFICIENTS** table this will update all lubricants in the sheet.
3. Copy data from .csv files to corresponding spots in the RAW COEFFICIENTS table.
4. Make comments in the comments section if you think it is necessary.
5. Now we must choose 2 sets of pressure matched data, for the first set typically choose low pressures (between 8-15 kPa). Do your best to choose data with the closest pressures.
6. Copy a pressure and its corresponding friction value from the RAW COEFFICIENTS table to the LOAD MATCHED COEFFICIENTS table for all of your repeats.
7. Repeat, choosing a higher pressure (typically around 20 kPa).



4.5 Recommended work order and data organization

Below is the recommended work order and way to organize your data. If you think of a more efficient way to do this please contact me (mlsamsom@gmail.com) to edit the analysis file and SOP or do so yourself.

1. Create appropriate files, it is recommended that the files are organized in the following way.
    a. A master file with your experiment number e.g. MS1411
    b. In the master folder create separate folders for repeats e.g. i, ii, iii etc.
    c. Place a copy of the excel summary file in the master folder.
    d. In each repeat folder create 4 folders: raw, notebooks, Docs, CSV.
2. Copy a notebook template in the notebook folder of the first repeat folder. e.g. EyeFrictionAnaysis1-3ipynb in \MS1411\i\notebooks
3. In the command line navigate to the notebook folder in the first repeat folder e.g. `michael@michael-Lenovo-G505s:~$ cd Dropbox\MS1411\i\notebooks`.
4. Start up the notebook either on your machine (`ipython notebook --ip=localhost`) or using the lab computer as a server.
5. Open the template notebook.
6. Update the General documentation section to correspond with your experiment.

7. Rename the notebook at the top to correspond with the first lubricant in your experiment (e.g. MS1411_PBS_i).
8. Run the notebook selecting the data for the first lubricant.
9. Change the name of the .csv output to correspond to your test e.g. MS1411_PBS_i.csv and move it to the CSV folder (it will spawn in the notebook folder).
10. copy your notebook, this will spawn a copy in a new tab of your browser.
    a. It is good practice to shutdown the previous notebook, this can be done in the IPython Dashboard tab in your browser (this isn't really an issue for this particular analysis but could create memory issues for more complex analyses).
11. rename the copy for the second lubricant e.g. MS1411_PRG4_i and update the pertinent fields in the General documentation section.
12. repeat steps 8-11 for each lubricant in the sequence.
13. Once one repeat has been completed, create PDF records of all your notebooks by typing or copying `ipython nbconvert --to latex --SphinxTransformer.author='Michael Samsom' --post PDF *.ipynb` in the command line. The current directory should still be the notebooks folder. Move all the .pdf files that will spawn in the notebook folder to the Docs folder. Delete any other files (can do this quickly by typing del *.tex in the command line).
14. Copy all the data from the .csv output files to the summary file using the above procedure.
15. Start on the next repeat by repeating steps 1-14 in the ii or subsequent folders (save time by copying a notebook from the previous analysis rather than reusing the template, be sure to update documentation though).

5. Troubleshooting

In the event of discrepancies in the data, there are some built in error tracking measures in the notebook file.

- The file analyzed is printed directly below the "Data IO and parsing" cell.
- This is the data for which the friction coefficients (printed right after "Data analysis" cell) and force traces (displayed at the end of the file).
- This was put in to track possible experimental and documentation errors.
- In addition to this, the ipython notebook environment gives a way to see exactly what code is being run for each file if errors are found in the code it is possible to track anywhere it was used. If you suspect there is an error in the analysis but are unfamiliar with the python programming language there are many resources online and tutorials on youtube. One such is http://www.afterhoursprogramming.com/tutorial/Python/Introduction/ but there are many

for every style of learner. Be sure to learn about numpy as it is used extensively in this analysis.

6. Figures for papers and presentations