# process_plate_reader_data.py documentation

Cody Martin*

June 14, 2021

| | |
|---|---|
| Author: | Cody Martin |
| Email: | ccmartin6@wisc.edu |
| Current Institution: | University of Wisconsin-Madison |
| Department: | Bacteriology |
| Notes: | Script created while affiliated with Dr. Ry Young lab at<br>Texas A&M University<br>Department of Biochemistry & Biophysics<br>Center for Phage Technology |
| Location: | https://github.com/cody-mar10/plate_reader.git |
| Version: | 2.0.0 |
| Date: | June 12, 2021 |

# Updates:

## Version 2.0.0

1. The script was given a major overhaul to make a more pythonic script. It is not far easier to troubleshoot, update, and read.
    - There are also more comments in the script for documentation.
    - A side effect of the more pythonic scripting is that the script runs much faster than previously. It shouldn't take longer than 15 seconds usually unless there is a large amount of data (full well plate used or >24 h time course). Even in those cases, it should not take more than a minute.
2. Added new arguments users can pass for more customization.
    - `-ts` to adjust time scale
    - `-o` to set the output directory
    - `-fd` to adjust the plot figure dimensions

*cody.martin@tamu.edu

3. Previously, if there were wells that contained data that were not meant to be included in the final data output, such as if they were for a separate experiment, users had to put ”air” in the plate setup for those wells. Now, users can just leave those well entries empty in the plate setup file.

4. I’ve also included my Jupyter Notebook if users want an interactive demonstration. It is also easier to customize settings for plotting too.

5. Previously, users has to specify the number of measurments taken per well at each timepoint, which defualts to 4 usually. Now, the script will automatically calculate this from the metadata in the sheet.

6. I’ve included an executable Rscript for plotting as well. If you would like to plot the data in a nicer R plot, you should first download R and run this command:

   - `./plot_platereader_data.R output/file`
   - where `output/file` is replaced by the file path to your long-formatted data

### What’s next

1. Because there are so many plotting parameters that could be customized, the best argument customization protocol would be to use a configuration file. I am working to implement this, so users would not need to pass an arguments on the command line. They would just need a config file with their desired parameters.

2. Need to determine a way to plot wide data if users specify wide-data (`excel`) output AND want the python plot.

3. Need to create special case where users have edited raw data and removed empty well tables.

## S1: INSTALL PYTHON and USE SHELL

1. You will need to be somewhat familiar with your computer system’s shell or command line tool.

   - For MacOS users, press: *CMD* + *SPACEBAR* and search ”Terminal”
   - For Windows users, press: START, type ”PowerShell”, and choose ”Windows PowerShell”

2. To run this script at the command line, you will need to have installed python on your computer. If you’re a python beginner, you can install miniconda3: `https://docs.conda.io/en/latest/miniconda.html`, which has bare minimum python packages. This code was tested in `python 3.9.4`.

3. You will then need to install 4 extra python packages:

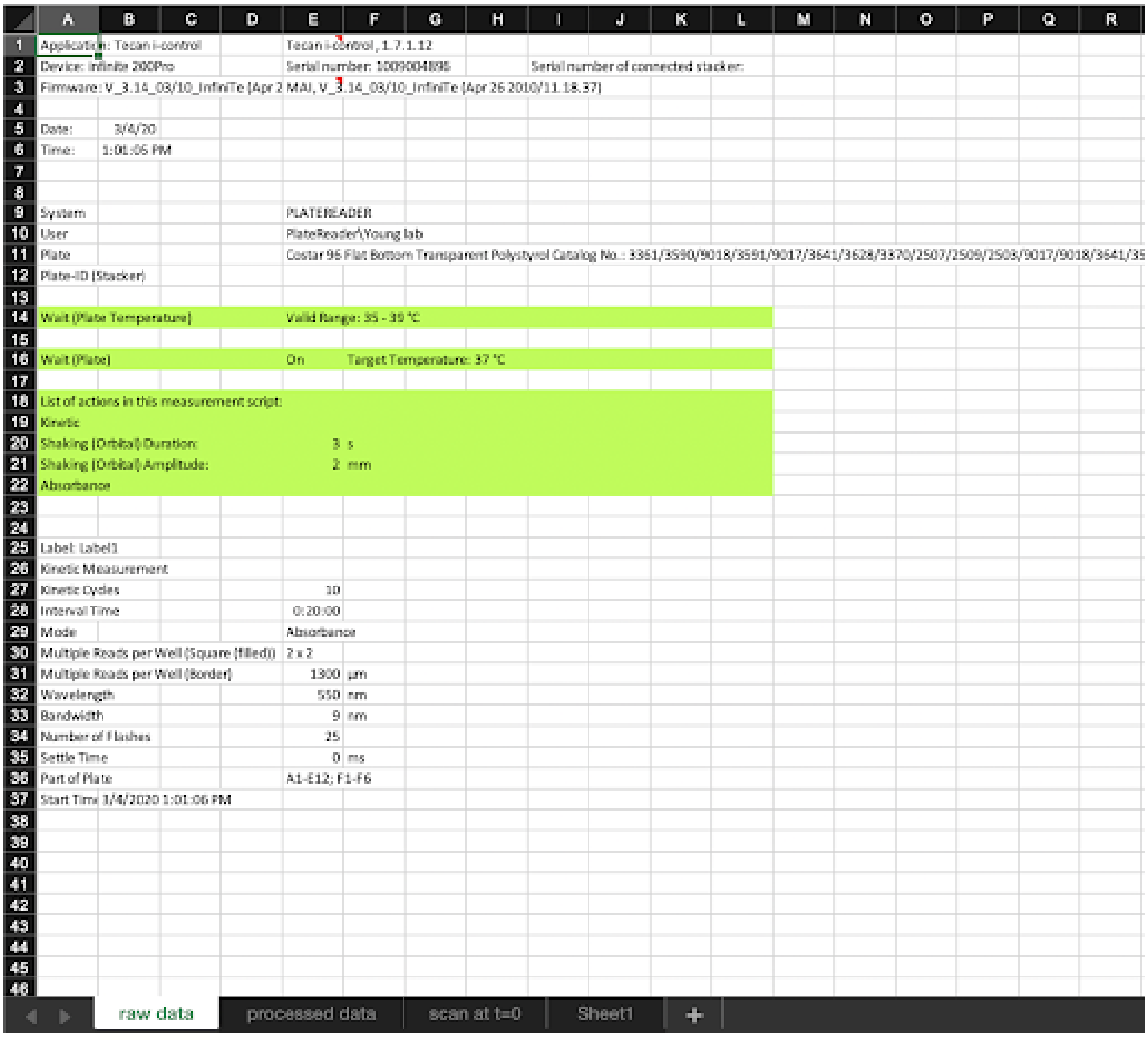
| package | version | build | command |
|---|---:|---|---|
| pandas | 1.1.3 | py38hb1e8313_0 | `conda install pandas=1.1.3` |
| openpyxl | 3.0.5 | py_0 | `conda install openpyxl=3.0.5` |
| matplotlib | 3.3.2 | 0 | `conda install matplotlib=3.3.2` |
| regex | 2020.10.15 | py39hb5aae12_0 | `conda install regex=2020.10.15` |

4. Then you can clone the repository to your local computer by using this command: `git clone https://github.com/cody-mar10/plate_reader.git`

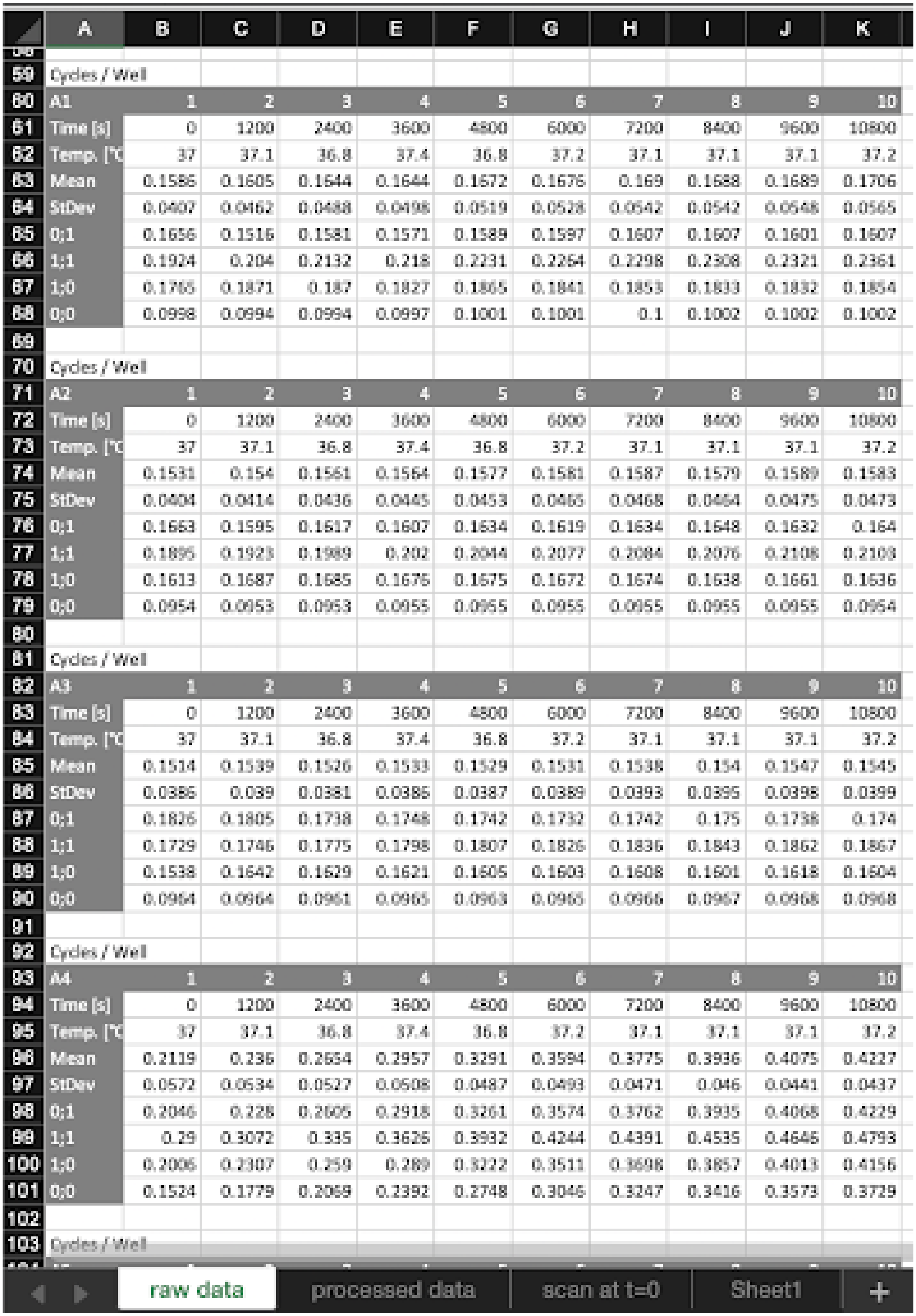
5. To change directories (folders) on MacOS/Linux, type: `cd ~/directory1/directory2/directory3/` until you are in your desired directory
   - NOTE: ”directory1,2,3” should be replaced with the actual folder name
   - For Windows users: I believe you can use the filepath as `C:\dir1\dir2\dir2`
6. When you clone this github repo, it will come with 2 test datasets and the plate setup keys for you to try out along with example outputs.
7. This repo can be anywhere on your device, but you will need the relative path to your data files needed as described in the next section.

## S2: INPUT FILES

1. You will need to input **two** files to run this code: your raw data and your plate setup file.
2. Your raw data must be the **unedited** .xlsx excel file that is output from the tecan i-control software. It should look very similar to this:



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Application: Tecan i-control | | | | Tecan i-control , 1.7.1.12 | | | | | | | | | | | | | |
| 2 | Device: infinite 200Pro | | | | Serial number: 1009004896 | | | | Serial number of connected stacker: | | | | | | | | | |
| 3 | Firmware: V_3.14_03/10_InfiniTe (Apr 2 | | | | MAI, V_3.14_03/10_InfiniTe (Apr 26 2010/11.18.37) | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | |
| 5 | Date: | 3/4/20 | | | | | | | | | | | | | | | | |
| 6 | Time: | 1:01:05 PM | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | |
| 9 | System | | | | PLATEREADER | | | | | | | | | | | | | |
| 10 | User | | | | PlateReader\Young lab | | | | | | | | | | | | | |
| 11 | Plate | | | | Costar 96 Flat Bottom Transparent Polystyrol Catalog No.: 3361/3590/9018/3591/9017/3641/3628/3370/2507/2509/2503/9017/9018/3641/35 | | | | | | | | | | | | | |
| 12 | Plate-ID (Stacker) | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | |
| 14 | Wait (Plate Temperature) | | | | Valid Range: 35 - 39 °C | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | |
| 16 | Wait (Plate) | | | | On | Target Temperature: 37 °C | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | |
| 18 | List of actions in this measurement script: | | | | | | | | | | | | | | | | | |
| 19 | Kinetic | | | | | | | | | | | | | | | | | |
| 20 | Shaking (Orbital) Duration: | | | | 3 | s | | | | | | | | | | | | |
| 21 | Shaking (Orbital) Amplitude: | | | | 2 | mm | | | | | | | | | | | | |
| 22 | Absorbance | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | |
| 25 | Label: Label1 | | | | | | | | | | | | | | | | | |
| 26 | Kinetic Measurement | | | | | | | | | | | | | | | | | |
| 27 | Kinetic Cycles | | | | 10 | | | | | | | | | | | | | |
| 28 | Interval Time | | | | 0:20:00 | | | | | | | | | | | | | |
| 29 | Mode | | | | Absorbance | | | | | | | | | | | | | |
| 30 | Multiple Reads per Well (Square (filled)) | | | | 2 x 2 | | | | | | | | | | | | | |
| 31 | Multiple Reads per Well (Border) | | | | 1300 | µm | | | | | | | | | | | | |
| 32 | Wavelength | | | | 550 | nm | | | | | | | | | | | | |
| 33 | Bandwidth | | | | 9 | nm | | | | | | | | | | | | |
| 34 | Number of Flashes | | | | 25 | | | | | | | | | | | | | |
| 35 | Settle Time | | | | 0 | ms | | | | | | | | | | | | |
| 36 | Part of Plate | | | | A1-E12; F1-F6 | | | | | | | | | | | | | |
| 37 | Start Tim | 3/4/2020 1:01:06 PM | | | | | | | | | | | | | | | | |
| 38 | | | | | | | | | | | | | | | | | | |
| 39 | | | | | | | | | | | | | | | | | | |
| 40 | | | | | | | | | | | | | | | | | | |
| 41 | | | | | | | | | | | | | | | | | | |
| 42 | | | | | | | | | | | | | | | | | | |
| 43 | | | | | | | | | | | | | | | | | | |
| 44 | | | | | | | | | | | | | | | | | | |
| 45 | | | | | | | | | | | | | | | | | | |
| 46 | | | | | | | | | | | | | | | | | | |



| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 59 | Cycles / Well | | | | | | | | | | |
| 60 | A1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 61 | Time [s] | 0 | 1200 | 2400 | 3600 | 4800 | 6000 | 7200 | 8400 | 9600 | 10800 |
| 62 | Temp. [°C | 37 | 37.1 | 36.8 | 37.4 | 36.8 | 37.2 | 37.1 | 37.1 | 37.1 | 37.2 |
| 63 | Mean | 0.1586 | 0.1605 | 0.1644 | 0.1644 | 0.1672 | 0.1676 | 0.169 | 0.1688 | 0.1689 | 0.1706 |
| 64 | StDev | 0.0407 | 0.0462 | 0.0488 | 0.0498 | 0.0519 | 0.0528 | 0.0542 | 0.0542 | 0.0548 | 0.0565 |
| 65 | 0;1 | 0.1656 | 0.1516 | 0.1581 | 0.1571 | 0.1589 | 0.1597 | 0.1607 | 0.1607 | 0.1601 | 0.1607 |
| 66 | 1;1 | 0.1924 | 0.204 | 0.2132 | 0.218 | 0.2231 | 0.2264 | 0.2298 | 0.2308 | 0.2321 | 0.2361 |
| 67 | 1;0 | 0.1765 | 0.1871 | 0.187 | 0.1827 | 0.1865 | 0.1841 | 0.1853 | 0.1833 | 0.1832 | 0.1854 |
| 68 | 0;0 | 0.0998 | 0.0994 | 0.0994 | 0.0997 | 0.1001 | 0.1001 | 0.1 | 0.1002 | 0.1002 | 0.1002 |
| 69 | | | | | | | | | | | |
| 70 | Cycles / Well | | | | | | | | | | |
| 71 | A2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 72 | Time [s] | 0 | 1200 | 2400 | 3600 | 4800 | 6000 | 7200 | 8400 | 9600 | 10800 |
| 73 | Temp. [°C | 37 | 37.1 | 36.8 | 37.4 | 36.8 | 37.2 | 37.1 | 37.1 | 37.1 | 37.2 |
| 74 | Mean | 0.1531 | 0.154 | 0.1561 | 0.1564 | 0.1577 | 0.1581 | 0.1587 | 0.1579 | 0.1589 | 0.1583 |
| 75 | StDev | 0.0404 | 0.0414 | 0.0436 | 0.0445 | 0.0453 | 0.0465 | 0.0468 | 0.0464 | 0.0475 | 0.0473 |
| 76 | 0;1 | 0.1663 | 0.1595 | 0.1617 | 0.1607 | 0.1634 | 0.1619 | 0.1634 | 0.1648 | 0.1632 | 0.164 |
| 77 | 1;1 | 0.1895 | 0.1923 | 0.1989 | 0.202 | 0.2044 | 0.2077 | 0.2084 | 0.2076 | 0.2108 | 0.2103 |
| 78 | 1;0 | 0.1613 | 0.1687 | 0.1685 | 0.1676 | 0.1675 | 0.1672 | 0.1674 | 0.1638 | 0.1661 | 0.1636 |
| 79 | 0;0 | 0.0954 | 0.0953 | 0.0953 | 0.0955 | 0.0955 | 0.0955 | 0.0955 | 0.0955 | 0.0955 | 0.0954 |
| 80 | | | | | | | | | | | |
| 81 | Cycles / Well | | | | | | | | | | |
| 82 | A3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 83 | Time [s] | 0 | 1200 | 2400 | 3600 | 4800 | 6000 | 7200 | 8400 | 9600 | 10800 |
| 84 | Temp. [°C | 37 | 37.1 | 36.8 | 37.4 | 36.8 | 37.2 | 37.1 | 37.1 | 37.1 | 37.2 |
| 85 | Mean | 0.1514 | 0.1539 | 0.1526 | 0.1533 | 0.1529 | 0.1531 | 0.1538 | 0.154 | 0.1547 | 0.1545 |
| 86 | StDev | 0.0386 | 0.039 | 0.0381 | 0.0386 | 0.0387 | 0.0389 | 0.0393 | 0.0395 | 0.0398 | 0.0399 |
| 87 | 0;1 | 0.1826 | 0.1805 | 0.1738 | 0.1748 | 0.1742 | 0.1732 | 0.1742 | 0.175 | 0.1738 | 0.174 |
| 88 | 1;1 | 0.1729 | 0.1746 | 0.1775 | 0.1798 | 0.1807 | 0.1826 | 0.1836 | 0.1843 | 0.1862 | 0.1867 |
| 89 | 1;0 | 0.1538 | 0.1642 | 0.1629 | 0.1621 | 0.1605 | 0.1603 | 0.1608 | 0.1601 | 0.1618 | 0.1604 |
| 90 | 0;0 | 0.0964 | 0.0964 | 0.0961 | 0.0965 | 0.0963 | 0.0965 | 0.0966 | 0.0967 | 0.0968 | 0.0968 |
| 91 | | | | | | | | | | | |
| 92 | Cycles / Well | | | | | | | | | | |
| 93 | A4 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 94 | Time [s] | 0 | 1200 | 2400 | 3600 | 4800 | 6000 | 7200 | 8400 | 9600 | 10800 |
| 95 | Temp. [°C | 37 | 37.1 | 36.8 | 37.4 | 36.8 | 37.2 | 37.1 | 37.1 | 37.1 | 37.2 |
| 96 | Mean | 0.2119 | 0.236 | 0.2654 | 0.2957 | 0.3291 | 0.3594 | 0.3775 | 0.3936 | 0.4075 | 0.4227 |
| 97 | StDev | 0.0572 | 0.0534 | 0.0527 | 0.0508 | 0.0487 | 0.0493 | 0.0471 | 0.046 | 0.0441 | 0.0437 |
| 98 | 0;1 | 0.2046 | 0.228 | 0.2605 | 0.2918 | 0.3261 | 0.3574 | 0.3762 | 0.3935 | 0.4068 | 0.4229 |
| 99 | 1;1 | 0.29 | 0.3072 | 0.335 | 0.3626 | 0.3932 | 0.4244 | 0.4391 | 0.4535 | 0.4646 | 0.4793 |
| 100 | 1;0 | 0.2006 | 0.2307 | 0.259 | 0.289 | 0.3222 | 0.3511 | 0.3698 | 0.3857 | 0.4013 | 0.4156 |
| 101 | 0;0 | 0.1524 | 0.1779 | 0.2069 | 0.2392 | 0.2748 | 0.3046 | 0.3247 | 0.3416 | 0.3573 | 0.3729 |
| 102 | | | | | | | | | | | |
| 103 | Cycles / Well | | | | | | | | | | |

- NOTE: your file name must not have spaces! Rename it to have no spaces (just use underscores _ ). This is good coding practice.
- It is **imperative** that you do not edit the raw data. If you remove the any of the tables for wells

that did not have samples, the script will fail. Version 2.0.0: I am working on a solution to this.

3. Your plate reader setup must be a .csv that looks like the example below.

   (a) You MUST have the headers ”row..1..2….12” AND the row letters in CAPS!

   (b) Name all replicates EXACTLY the same (just copy and paste)

   - **Version 2.0.0 Update**: Your sample names can be anything EXCEPT for ”All”.

   (c) **Label your media-only wells as Blank**

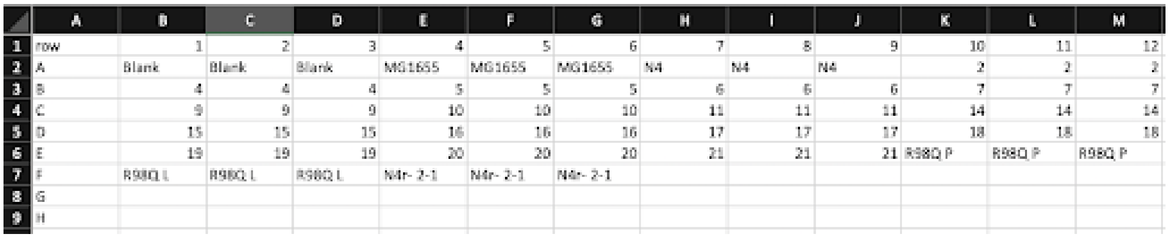
   (d) Leave empty wells blank

   - **Version 2.0.0 Update**: If you have any wells that were scanned, but you do not want to include the data in the final output, just leave those wells blank in the plate setup file. The code will automatically ignore them.

   (e) Make sure there are not any extra spaces anywhere

   (f) You can make a .csv file in excel. Just save it as .csv

   - Your sample names can now be whatever you want, as long as replicates are named EXACTLY the same.
   - However, I would suggest you use actual sample names instead of a code.



|   | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | row | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 2 | A | Blank | Blank | Blank | MG1655 | MG1655 | MG1655 | N4 | N4 | N4 | 2 | 2 | 2 |
| 3 | B | 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 |
| 4 | C | 9 | 9 | 9 | 10 | 10 | 10 | 11 | 11 | 11 | 14 | 14 | 14 |
| 5 | D | 15 | 15 | 15 | 16 | 16 | 16 | 17 | 17 | 17 | 18 | 18 | 18 |
| 6 | E | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | R98Q P | R98Q P | R98Q P |
| 7 | F | R98Q L | R98Q L | R98Q L | N4r- 2-1 | N4r- 2-1 | N4r- 2-1 |  |  |  |  |  |  |
| 8 | G |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | H |  |  |  |  |  |  |  |  |  |  |  |  |

# S3: USAGE

1. At the command line, once the script and your input files are in the same directory, you can use this code like this:

```
python process_plate_reader_data.py −i <rawdata.xlsx> −p <plate_setup.
   csv> −a <active_sheet_num>
```

2. A complete command could look like this:

```
python process_plate_reader_data.py −i 200221
   _CCM_n4_lysis_curve_testing.xlsx −p 200221_plate_setup.csv −a 1
```

3. There are several arguments that can be input when you call the script, but the 3 that are most important are -i, -p, and -a.

4. -i specifies the input data file

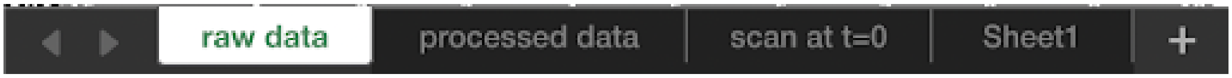
5. -p specifies the plate reader setup

6. -a specifies the sheet from your input data that has the raw data

7. **Note: computers count from 0**, so down below, I would enter 0 (actually you wouldn't have to do anything because the default setting is the first sheet). If the second sheet had the data, I would enter

(a)

8. Additionally, for help you can pass -h as your argument to get the list of possible args



Sheet = 0 Sheet = 1 Sheet = 2 Sheet = 3

9. Here are all the possible arguments you can pass and their descriptions:

| Arg | Description |
|---|---|
| -i | input must be a .xlsx excel file from infinite200 pro plate reader tecan i-control 1.7.1.12 software |
| -p | plate setup file must be structured a certain way as a .csv – see the example<br>NOTE: replicates must be named exactly the same AND your background media samples called ”Blank”<br>Should you have samples you don’t want analyzed or wells that were scanned but you don’t want analyzed, label them as ”air” |
| -a | type an integere for the sheet in your .xlsx file with the raw data<br>NOTE: computers start counting at 0 so if your 2nd sheet has the data, type ”1”<br>DEFAULT=0 (first sheet) |
| -gs | type your intended graphing program<br>options: ”excel”, ”R” DEFAULT=”R”<br>”excel” will output wide formatted data<br>”R” will output long formatted data for use in R with ggplot2 |
| -pp | bool to autogenerate a python plot<br>DEFAULT=True<br>Note: To set this to false, you need to pass `-pp` on the command line with nothing else. The script will automatically set this to False.<br>**Version 2.0.0** To produce a plot, the data must be long formatted, ie you must have the `-gs` parameter `R`. |
| -y | y-axis label<br>DEFAULT=”OD600” |
| -gm | type of graphing method for python created graph<br>options: ”time series”, ”time diff” DEFAULT=”time series”<br>NOTE passing ”time diff” does nothing yet TODO |
| -pw | type an integer for number of points (readings) per well<br>DEFAULT=4<br>**Version 2.0.0**: Deprecated. This is calculated from metadata in excel file. |
| -ts | **Version 2.0.0** Time scale. Determines what time scale is output in both the returned processed data and plots<br>Options: ”sec”, ”min”, ”hr”<br>Default=”hr”. |
| -o | **Version 2.0.0** Output directory. Specify where you want the output files.<br>Default=”output” |
| -fd | **Version 2.0.0** Figure dimensions. When the python plot is produced, specify the dimensions of the plot in inches.<br>Default=(10,10)<br>Usage: To change this parameter, type `(w, h)` WITH the quotes. |

# S4: HOW DOES IT WORK

1. The code takes your plate setup file, and groups all replicates together. Then it will find the well coordinates for all data in your plate, grouping the coordinates for replicates together.

2. Then it parses all the data tables in your raw data file and matches the data table with a sample labeled in your plate setup file.

3. Then it will concatenate all data tables together for all replicates of a single experimental group. In other words, if you have 3 replicates of ”rapid_lyser_2” in wells B2, B3, and B4, it will take the data for wells B2, 3, and 4, and create one data table of all 3 replicates for ”rapid_lyser_2.”

4. Then it takes the average and standard deviation of all data points for a given experimental group at each timepoint. For the tecan i-control software, the data includes a certain number of readings per well, and then that number is multiplied by the number of replicates. The default readings per well number is 4, and if you have 3 replicates, that equals 12 data points for each group at each timepoint.

5. The average background signal at each timepoint is then subtractedOo from the average signal at the corresponding timepoint for each experimental group. This value is what is present in the final output. Additionally, the unchanged standard deviation is also reported in the final output because it will not change by subtracting background signal. (See statistic distribution theory)

6. For the time series plotting, it will plot the relative time (hours) on the x-axis, and whatever units your measurements are in on the y-axis. For each point, it will also graph the standard error of the mean OD ± SEM. at that timepoint. SEM = standard dev / sqrt(n), where n is the number of replicates * number of readings per well.

7. The outputs of this code are a `datafile_PROCESSED.csv` file and a `datafile_PROCESSED.pdf` file. The .csv file contains for all experimental groups, the average signal (with background subtracted) and standard deviation at each timepoint, as well as the time in hours (by default). The .pdf is a python plot.