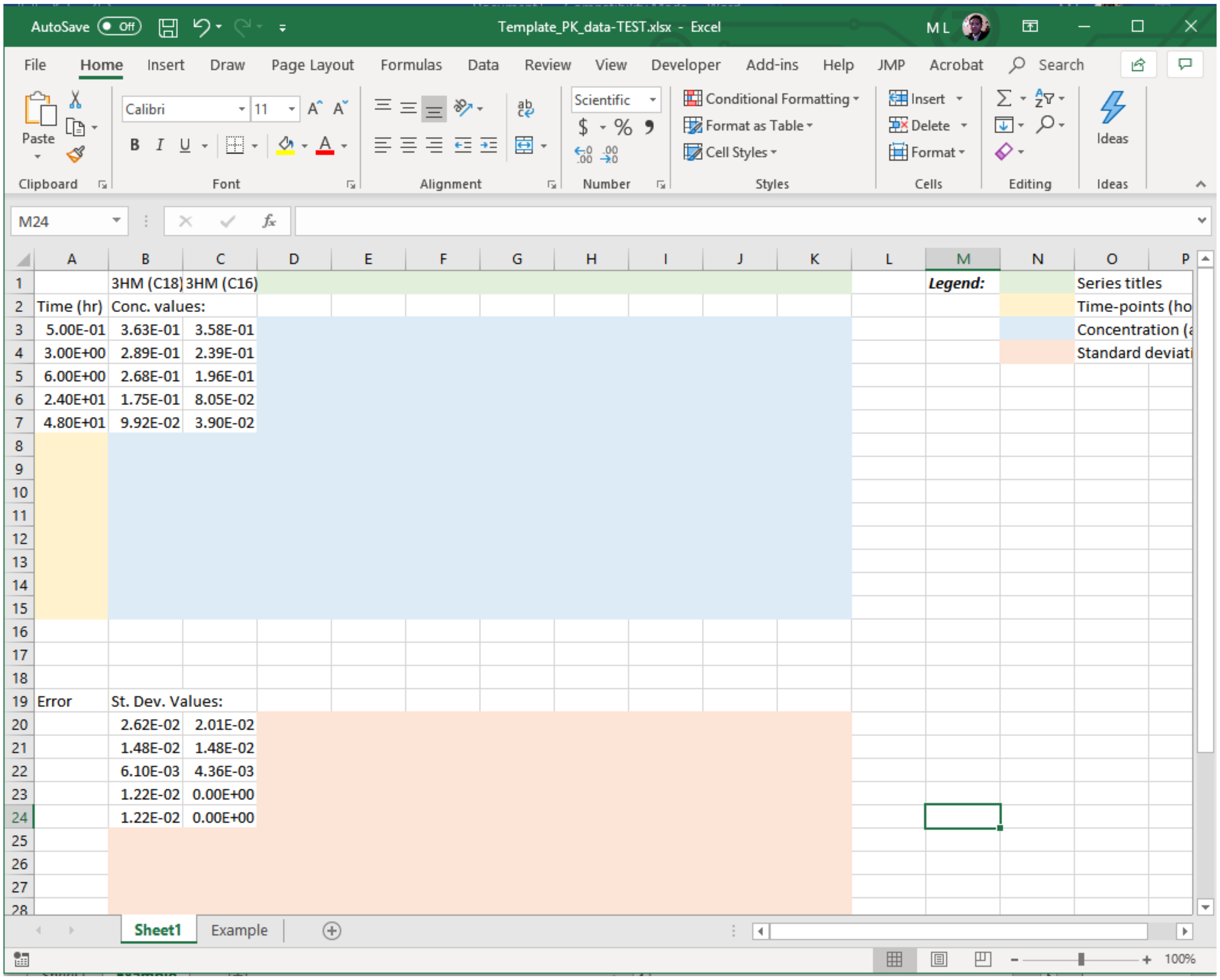
Running DFM-Fitting Code in JupyterLab Online Demo

1. Download “DFM Jupyter NB.ipynb”, “Template_PK_data.xlsx”, and “Template_TUMOR_data.xlsx” files from https://github.com/mlim0789/DFM-Fitting

2. We will verify the code using example data. In the template PK and Tumor data files, copy and paste values from Example Sheet to Sheet 1. Save files.

Excel sheet after copying and pasting Example PK values:

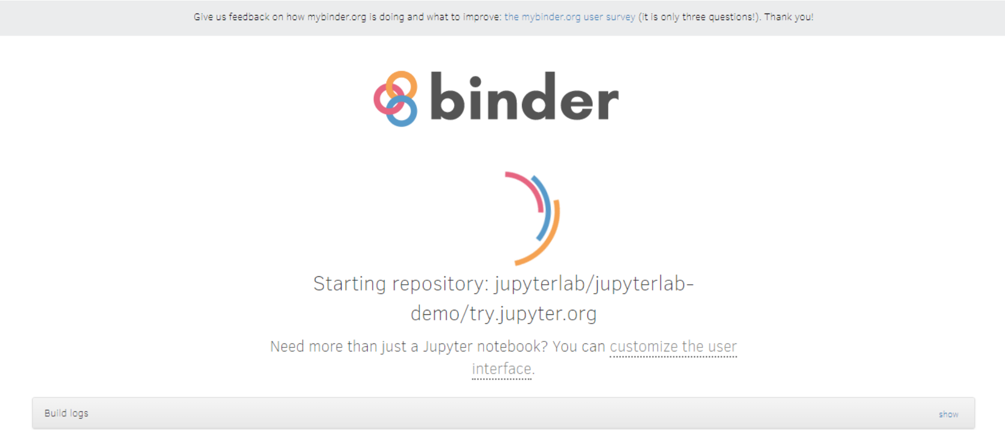


	A	B	C	M	N	O
1		3HM (C18)	3HM (C16)	*Legend:*	🟩	Series titles
2	Time (hr)	Conc. values:			🟨	Time-points (ho
3	5.00E-01	3.63E-01	3.58E-01		🟦	Concentration (a
4	3.00E+00	2.89E-01	2.39E-01		🟧	Standard deviati
5	6.00E+00	2.68E-01	1.96E-01			
6	2.40E+01	1.75E-01	8.05E-02			
7	4.80E+01	9.92E-02	3.90E-02			
19	Error	St. Dev. Values:				
20		2.62E-02	2.01E-02			
21		1.48E-02	1.48E-02			
22		6.10E-03	4.36E-03			
23		1.22E-02	0.00E+00			
24		1.22E-02	0.00E+00			

3. Click on the following link to launch JupyterLab demo:

https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/try.jupyter.org?urlpath=lab

4. The loading page will appear as the interactive development environment is being set up



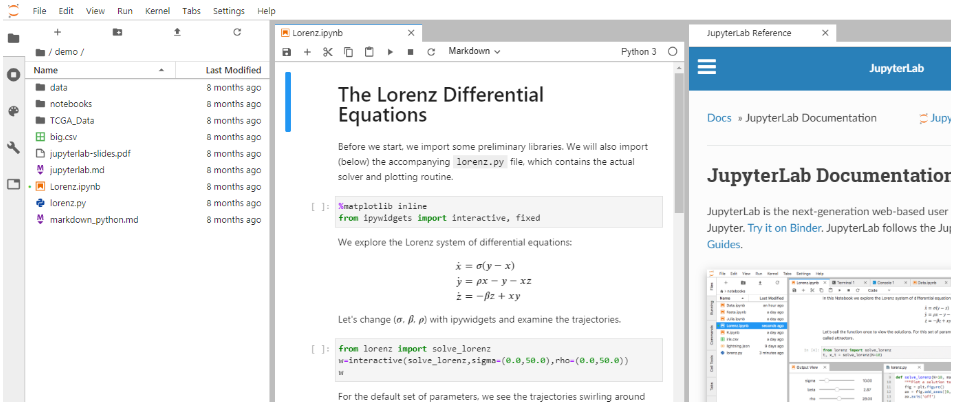
Give us feedback on how mybinder.org is doing and what to improve: the mybinder.org user survey (it is only three questions!). Thank you!

Starting repository: jupyterlab/jupyterlab-demo/try.jupyter.org

Need more than just a Jupyter notebook? You can customize the user interface.

5. You may see a demo notebook appear, or just the launcher landing page

Demo notebook:



Name	Last Modified
data	8 months ago
notebooks	8 months ago
TCGA_Data	8 months ago
big.csv	8 months ago
jupyterlab-slides.pdf	8 months ago
jupyterlab.md	8 months ago
Lorenz.ipynb	8 months ago
lorenz.py	8 months ago
markdown_python.md	8 months ago

The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
[ ]: %matplotlib inline
     from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= \rho x - y - xz\\ \dot{z} &= -\beta z + xy\end{aligned}$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
[ ]: from lorenz import solve_lorenz
     w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
     w
```

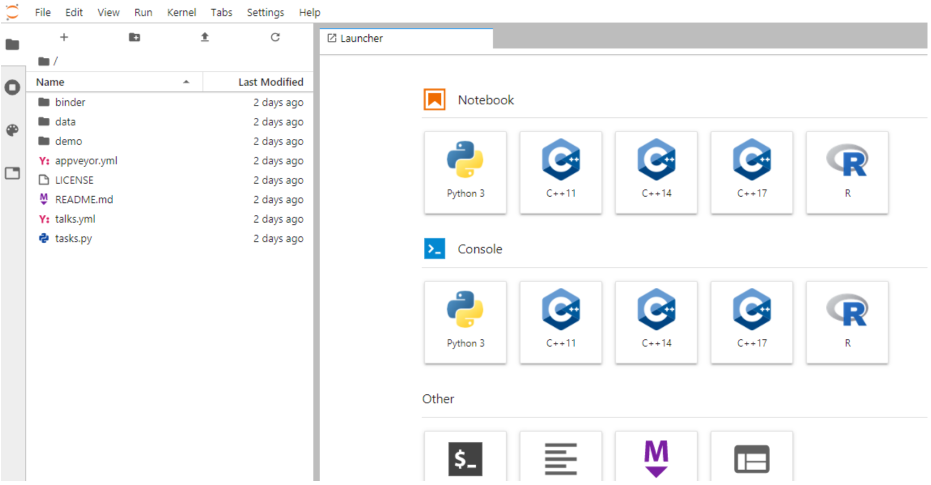
For the default set of parameters, we see the trajectories swirling around

Docs » JupyterLab Documentation

JupyterLab Documentation

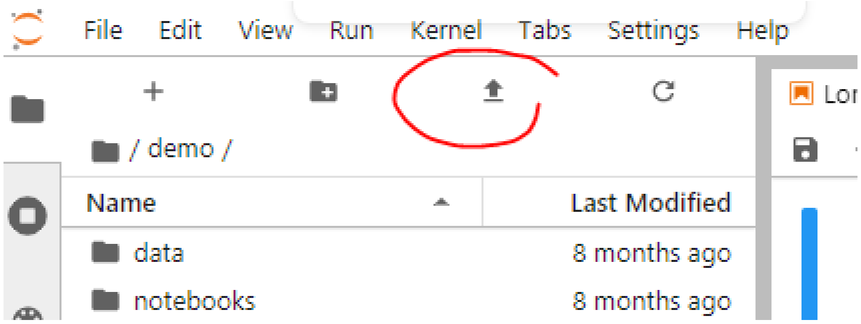
JupyterLab is the next-generation web-based user
Jupyter. Try it on Binder. JupyterLab follows the Ju
Guides.

Launcher page:



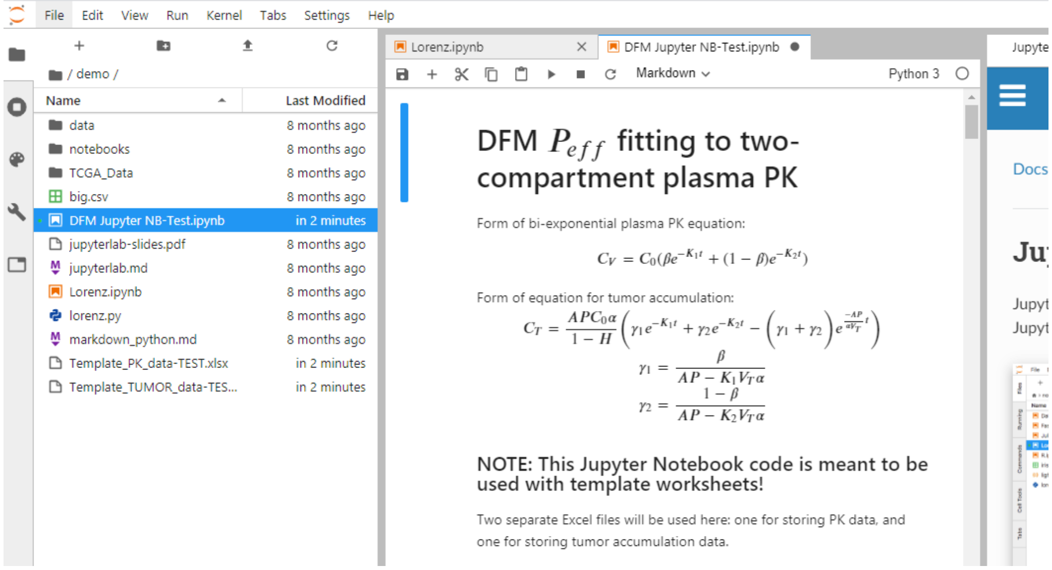
Name	Last Modified
binder	2 days ago
data	2 days ago
demo	2 days ago
appveyor.yml	2 days ago
LICENSE	2 days ago
README.md	2 days ago
talks.yml	2 days ago
tasks.py	2 days ago

6. Either way, upload your files by clicking on the arrow icon:



Name	Last Modified
data	8 months ago
notebooks	8 months ago

7. After uploading the JupyterLab notebook, and PK & TUMOR data files, they will appear on the left panel. Click on the "DFM Jupyter NB" file:


File Edit View Run Kernel Tabs Settings Help
/ demo /
Lorenz.ipynb
DFM Jupyter NB-Test.ipynb
Markdown
Python 3
Jupyte
Docs
Ju
Jupyt
Jupyt

Name	Last Modified
data	8 months ago
notebooks	8 months ago
TCGA_Data	8 months ago
big.csv	8 months ago
DFM Jupyter NB-Test.ipynb	in 2 minutes
jupyterlab-slides.pdf	8 months ago
jupyterlab.md	8 months ago
Lorenz.ipynb	8 months ago
lorenz.py	8 months ago
markdown_python.md	8 months ago
Template_PK_data-TEST.xlsx	in 2 minutes
Template_TUMOR_data-TES...	in 2 minutes

DFM P_{eff} fitting to two-compartment plasma PK

Form of bi-exponential plasma PK equation:

$$C_V = C_0(\beta e^{-K_1 t} + (1-\beta)e^{-K_2 t})$$

Form of equation for tumor accumulation:

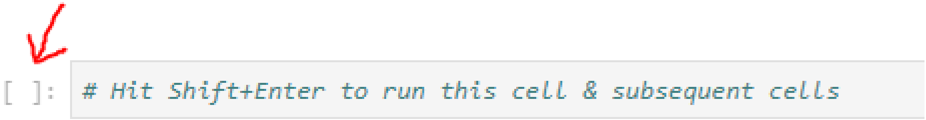
$$C_T = \frac{APC_0\alpha}{1-H}\left(\gamma_1 e^{-K_1 t} + \gamma_2 e^{-K_2 t} - \left(\gamma_1 + \gamma_2\right) e^{\frac{-AP}{\alpha V_T} t}\right)$$
$$\gamma_1 = \frac{\beta}{AP - K_1 V_T \alpha}$$
$$\gamma_2 = \frac{1-\beta}{AP - K_2 V_T \alpha}$$

NOTE: This Jupyter Notebook code is meant to be used with template worksheets!

Two separate Excel files will be used here: one for storing PK data, and one for storing tumor accumulation data.

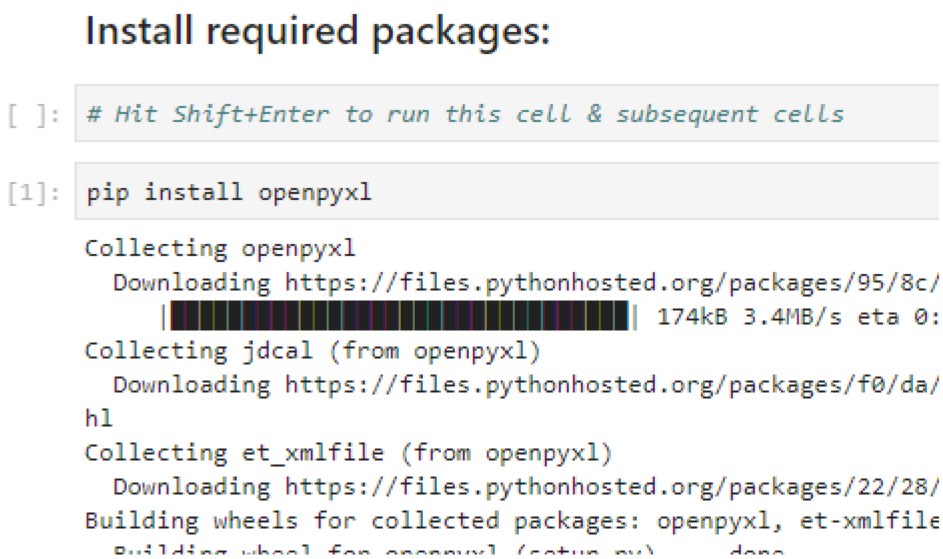
8. When you scroll down, you'll see grey boxes, which are cells containing executable codes. Hitting Shift+Enter will run the code.

   Square brackets on the left side indicate the status of running the code. When it is empty, it has not been run. While running, it will display "[*]", and when it is finished, a number pops up. Any outputs will be shown below the cell.



```
[ ]: # Hit Shift+Enter to run this cell & subsequent cells
```

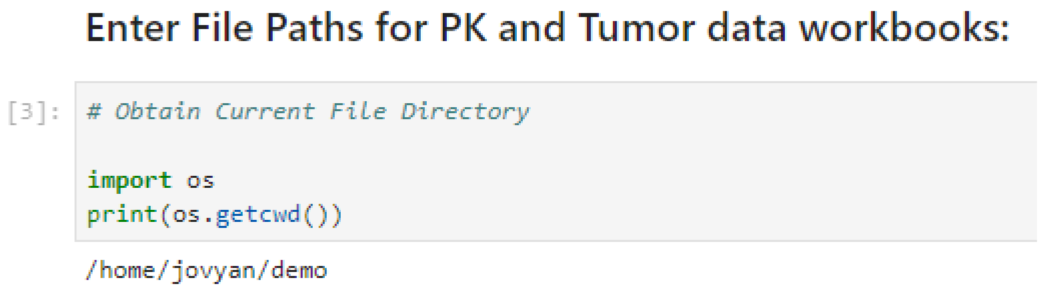
9. First, there are code packages that need to be installed. Run the command for installing openpyxl and tabulate, and you should see an output like this from each command:



Install required packages:

```
[ ]: # Hit Shift+Enter to run this cell & subsequent cells

[1]: pip install openpyxl

Collecting openpyxl
  Downloading https://files.pythonhosted.org/packages/95/8c/
     |████████████████████████████████| 174kB 3.4MB/s eta 0:
Collecting jdcal (from openpyxl)
  Downloading https://files.pythonhosted.org/packages/f0/da/
hl
Collecting et_xmlfile (from openpyxl)
  Downloading https://files.pythonhosted.org/packages/22/28/
Building wheels for collected packages: openpyxl, et-xmlfile
  Building wheel for openpyxl (setup.py) ... done
```

10. Next, enter file paths for the PK and tumor data. You'll need the current file directory. For this JupyterLab demo, it'll likely be "/home/jovyan/demo"

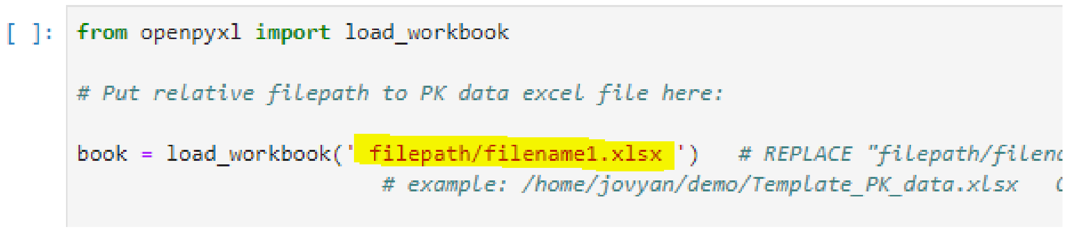


Enter File Paths for PK and Tumor data workbooks:

```
[3]: # Obtain Current File Directory

     import os
     print(os.getcwd())

     /home/jovyan/demo
```

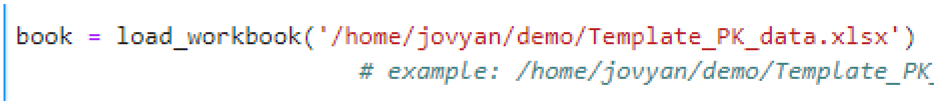
11. Replace the filepaths for load_workbook command accordingly. Ensure there are **no excess spaces** between the single quotation marks at the end!

Before replacing filepath:



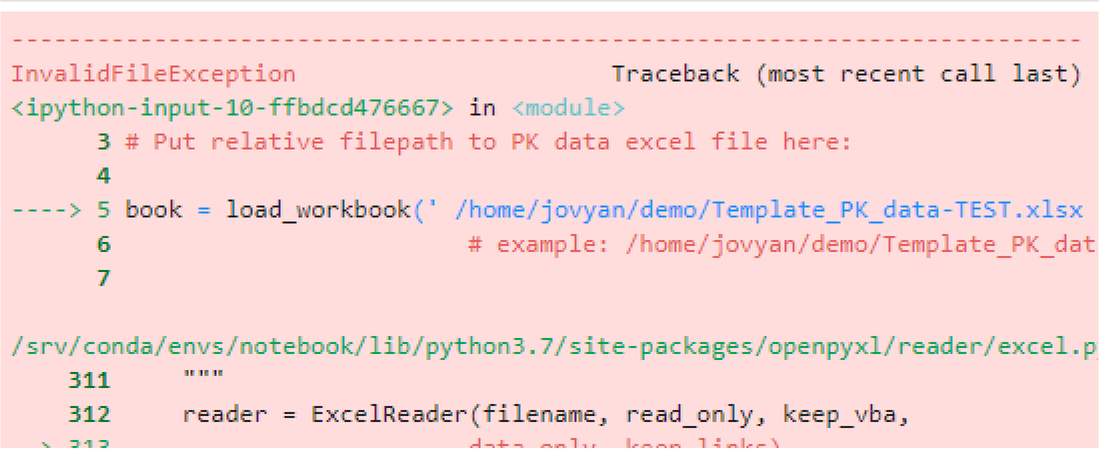
```
[ ]: from openpyxl import load_workbook

     # Put relative filepath to PK data excel file here:

     book = load_workbook(' filepath/filename1.xlsx ')    # REPLACE "filepath/filenc
                                # example: /home/jovyan/demo/Template_PK_data.xlsx   (
```

E.g. After replacing filepath:



```
book = load_workbook('/home/jovyan/demo/Template_PK_data.xlsx')
                          # example: /home/jovyan/demo/Template_PK_
```

12. If you see an error like these, double check that there are no excess spaces between the filepath quotes, and that the file name has been entered correctly

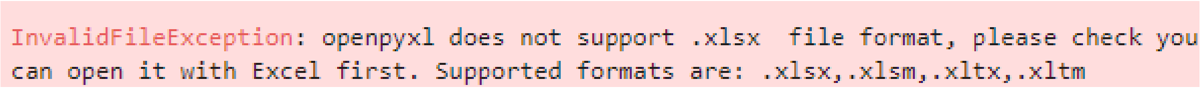


```
---------------------------------------------------------------------------
InvalidFileException                      Traceback (most recent call last)
<ipython-input-10-ffbdcd476667> in <module>
      3 # Put relative filepath to PK data excel file here:
      4 
----> 5 book = load_workbook(' /home/jovyan/demo/Template_PK_data-TEST.xlsx
      6                         # example: /home/jovyan/demo/Template_PK_dat
      7 

/srv/conda/envs/notebook/lib/python3.7/site-packages/openpyxl/reader/excel.p
    311     """
    312     reader = ExcelReader(filename, read_only, keep_vba,
--> 313                         data_only, keep_links)
```

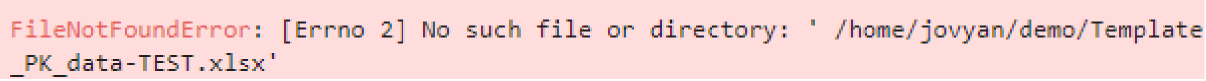
.

.

.

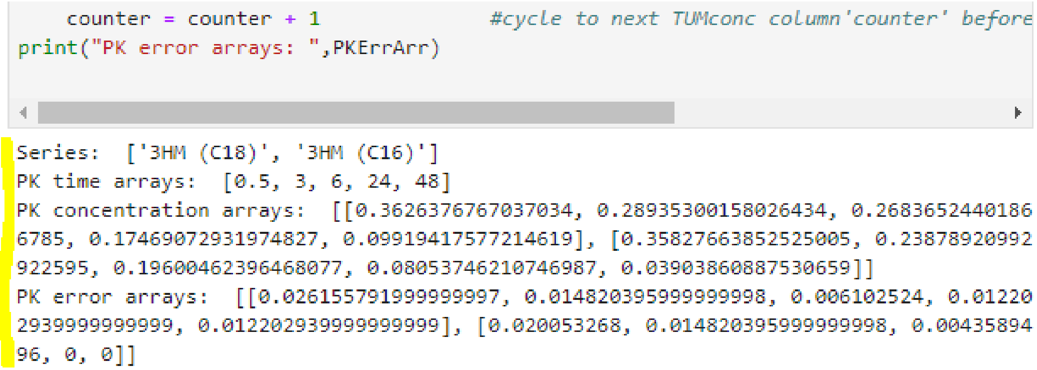


```
InvalidFileException: openpyxl does not support .xlsx  file format, please check you
can open it with Excel first. Supported formats are: .xlsx,.xlsm,.xltx,.xltm
```



```
FileNotFoundError: [Errno 2] No such file or directory: ' /home/jovyan/demo/Template
_PK_data-TEST.xlsx'
```

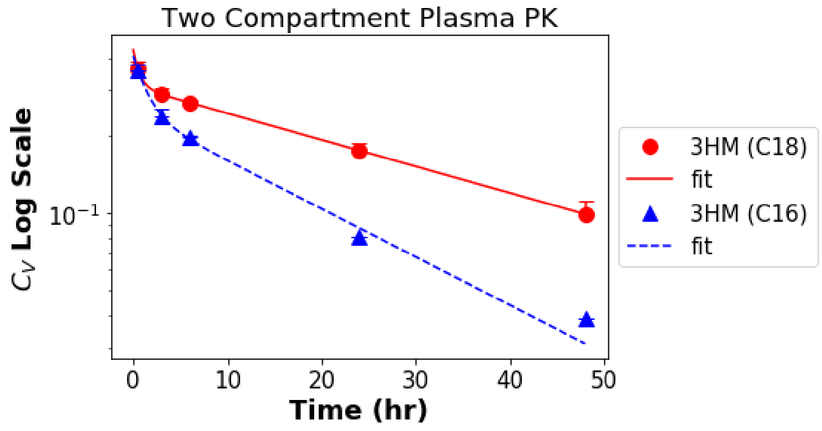
13. Run the commands for verifying PK and tumor data loading. You should see an output like this, which should match values on the Excel file.



```
    counter = counter + 1          #cycle to next TUMconc column'counter' before
print("PK error arrays: ",PKErrArr)
```

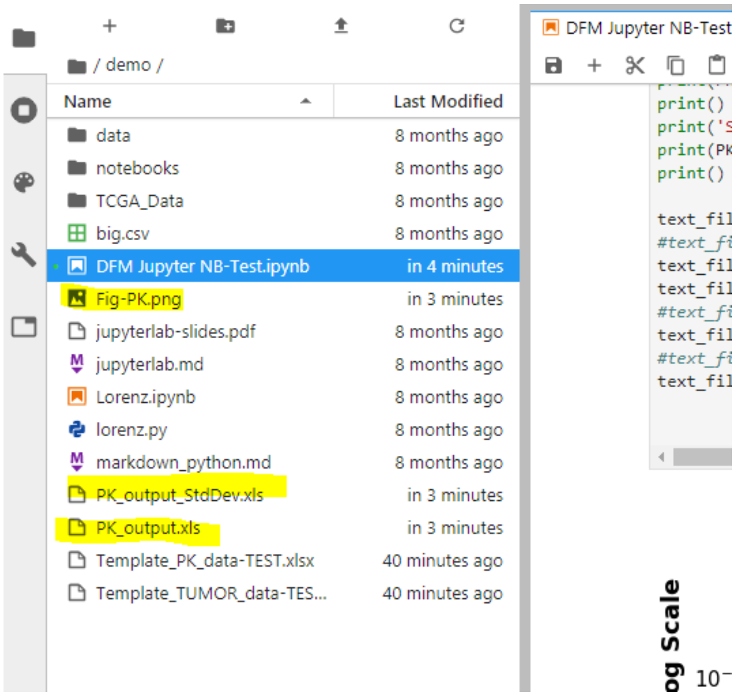
```
Series:  ['3HM (C18)', '3HM (C16)']
PK time arrays:  [0.5, 3, 6, 24, 48]
PK concentration arrays:  [[0.3626376767037034, 0.28935300158026434, 0.2683652440186
6785, 0.17469072931974827, 0.09919417577214619], [0.35827663852525005, 0.23878920992
922595, 0.19600462396468077, 0.08053746210746987, 0.03903860887530659]]
PK error arrays:  [[0.026155791999999997, 0.014820395999999998, 0.006102524, 0.01220
2939999999999, 0.012202939999999999], [0.020053268, 0.014820395999999998, 0.00435894
96, 0, 0]]
```

14. Run the code for fitting PK data, and you should see an output that looks like this



```
        3HM (C18)    3HM (C16)
----  -----------  -----------
C0      0.43042      0.407854
beta    0.281396     0.397448
K1      1.50796      0.640804
K2      0.0237371    0.0430696
R^2     0.999997     0.998278

Standard deviations:
        3HM (C18)    3HM (C16)
----  -----------  -----------
C0    0.0100995    0.0295186
beta  0.0159962    0.0572274
K1    0.172918     0.384638
K2    8.54976e-05  0.00680832
```

15. Note that the graph and table output values have been automatically saved as PNG image (Fig-PK.png) and .xls worksheet files, available on the left panel! You can open and download these.

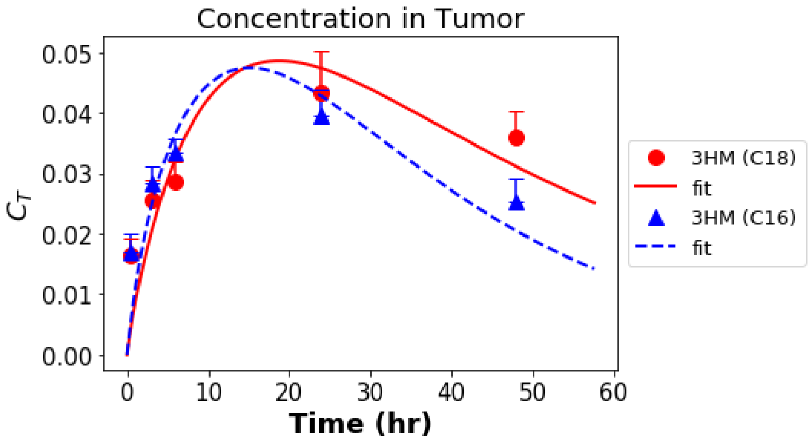


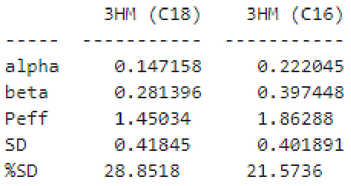
Name	Last Modified
data	8 months ago
notebooks	8 months ago
TCGA_Data	8 months ago
big.csv	8 months ago
DFM Jupyter NB-Test.ipynb	in 4 minutes
Fig-PK.png	in 3 minutes
jupyterlab-slides.pdf	8 months ago
jupyterlab.md	8 months ago
Lorenz.ipynb	8 months ago
lorenz.py	8 months ago
markdown_python.md	8 months ago
PK_output_StdDev.xls	in 3 minutes
PK_output.xls	in 3 minutes
Template_PK_data-TEST.xlsx	40 minutes ago
Template_TUMOR_data-TES...	40 minutes ago

```
print()
print('S
print(PK
print()

text_fil
#text_fi
text_fil
text_fil
#text_fi
text_fil
#text_fi
text_fil
```

16. Now that the PK values have been successfully fitted, we can perform the actual DFM fit on tumor accumulation data.

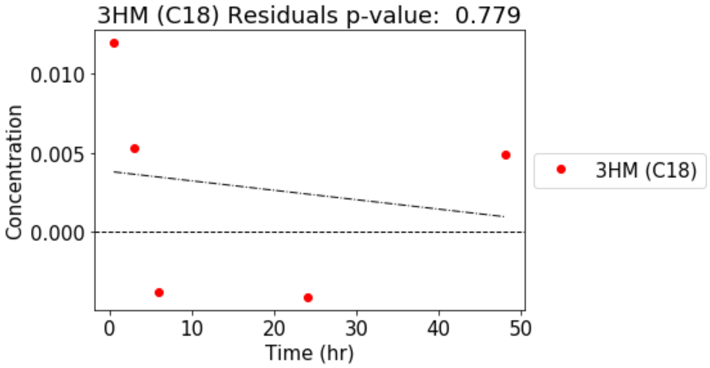
    Run the code for DFM Fitting. You should see the following output, with figure and table files saved to the left panel accordingly. Effective permeability data table is saved as "Peff_output.xls"

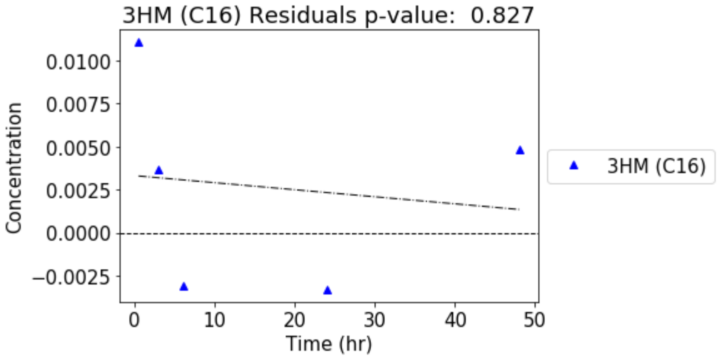




```
             3HM (C18)      3HM (C16)
-----     -----------    -----------
alpha        0.147158       0.222045
beta         0.281396       0.397448
Peff         1.45034        1.86288
SD           0.41845        0.401891
%SD         28.8518        21.5736
```

17. Finally, run the code to generate residuals of the fit for each series





18. If you see all of the above outputs, congratulations, the DFM fitting code has run successfully!

19. Next, replace the PK and Tumor data with your values in the Excel template Sheet 1, and repeat the PK & DFM fitting process to extract effective permeability values for your system.

Thank you for your interest in our work. We sincerely hope that this method serves as a valuable tool to complement your cancer nanomedicine research. If you have any questions, comments, or suggestions, please direct them to Marc Lim at mlim0789[at]gmail.com (ResearchGate, LinkedIn); or Professor Ting Xu at tingxu[at]berkeley.edu (UCB Faculty profile).