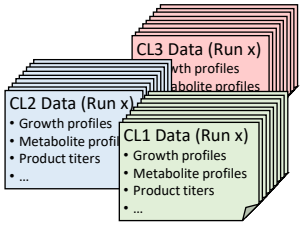
Cell Culture Data Pipeline Python Package (CDApy)



CDApy
- Calculation
- Aggregation
- Visualization

Aggregated_Data.csv

Process Data Entry (Excel)

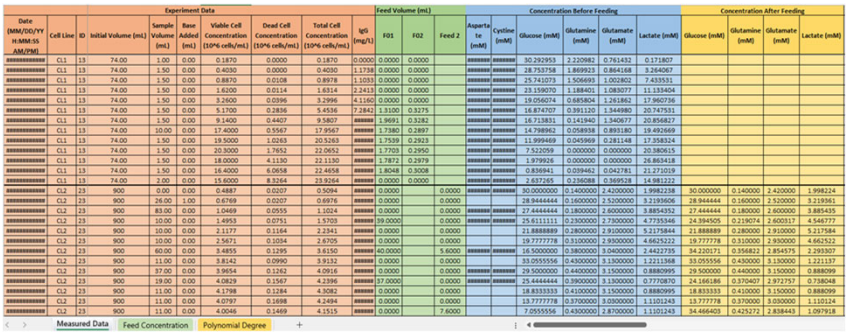
Multiple batches and cell lines

Additional features compared to Excel worksheet:
- Process multiple batches and cell lines at the same time
- Provide rolling window regression for specific-rate calculation
- Provide interactive plotting to analyze and compare process profiles

1

1

Excel data input (1): process data entry



Experiment Data										Feed Volume (mL)			Concentration Before Feeding						Concentration After Feeding			
Date (MM/DD/YY H:MM:SS AM/PM)	Cell Line	ID	Initial Volume (mL)	Sample Volume (mL)	Base Added (mL)	Viable Cell Concentration (10^6 cells/mL)	Dead Cell Concentration (10^6 cells/mL)	Total Cell Concentration (10^6 cells/mL)	IgG (mg/L)	F01	F02	Feed 2	Aspartate (mM)	Cystine (mM)	Glucose (mM)	Glutamine (mM)	Glutamate (mM)	Lactate (mM)	Glucose (mM)	Glutamine (mM)	Glutamate (mM)	Lactate (mM)
##########	CL1	13	74.00	1.00	0.00	0.1870	0.0000	0.1870	0.0000	0.0000	0.0000		########	########	30.292953	2.220982	0.761432	0.171807				
##########	CL1	13	74.00	1.50	0.00	0.4030	0.0000	0.4030	1.1738	0.0000	0.0000		########	########	28.753758	1.869923	0.864168	3.264067				
##########	CL1	13	74.00	1.50	0.00	0.8870	0.0108	0.8978	1.1033	0.0000	0.0000		########	########	25.741073	1.506693	1.002802	7.433531				
##########	CL1	13	74.00	1.50	0.00	1.6200	0.0114	1.6314	2.2413	0.0000	0.0000		########	########	23.159070	1.188401	1.083077	11.133404				
##########	CL1	13	74.00	1.50	0.00	3.2600	0.0396	3.2996	4.1160	0.0000	0.0000		########	########	19.056074	0.685804	1.261862	17.960736				
##########	CL1	13	74.00	1.50	0.00	5.1700	0.2836	5.4536	7.2842	1.3100	0.3275		########	########	16.874707	0.391120	1.344980	20.747531				
##########	CL1	13	74.00	1.50	0.00	9.1400	0.4407	9.5807	######	1.9691	0.3282		########	########	16.713831	0.141940	1.340677	20.856827				
##########	CL1	13	74.00	10.00	0.00	17.4000	0.5567	17.9567	######	1.7380	0.2897		########	########	14.798962	0.058938	0.893180	19.492669				
##########	CL1	13	74.00	1.50	0.00	19.5000	1.0263	20.5263	######	1.7539	0.2923		########	########	11.999469	0.045969	0.281148	17.358324				
##########	CL1	13	74.00	1.50	0.00	20.3000	1.7652	22.0652	######	1.7703	0.2950		########	########	7.522059	0.000000	0.000000	20.380615				
##########	CL1	13	74.00	1.50	0.00	18.0000	4.1150	22.1150	######	1.7872	0.2979		########	########	1.979926	0.000000	0.000000	26.863418				
##########	CL1	13	74.00	1.50	0.00	16.4000	6.0658	22.4658	######	1.8048	0.3008		########	########	0.836941	0.039462	0.042781	21.271019				
##########	CL1	13	74.00	2.00	0.00	15.6000	8.3264	23.9264	######	0.0000	0.0000		########	########	2.637265	0.236088	0.369528	14.981222				
##########	CL2	23	900	0.00	0.00	0.4887	0.0207	0.5094	######	0.0000		0.0000	########	########	30.0000000	0.1400000	2.4200000	1.9982238	30.000000	0.140000	2.420000	1.998224
##########	CL2	23	900	26.00	1.00	0.6769	0.0207	0.6976	######	0.0000		0.0000			28.9444444	0.1600000	2.5200000	3.2193606	28.944444	0.160000	2.520000	3.219361
##########	CL2	23	900	83.00	0.00	1.0469	0.0555	1.1024	######	0.0000		0.0000	########	########	27.4444444	0.1800000	2.6000000	3.8854352	27.444444	0.180000	2.600000	3.885435
##########	CL2	23	900	10.00	0.00	1.4953	0.0751	1.5703	######	39.0000		0.0000	########	########	25.6111111	0.2300000	2.7300000	4.7735346	24.394505	0.219074	2.600317	4.546777
##########	CL2	23	900	10.00	0.00	2.1177	0.1164	2.2341	######	0.0000		0.0000			21.8888889	0.2800000	2.9100000	5.2175844	21.888889	0.280000	2.910000	5.217584
##########	CL2	23	900	10.00	0.00	2.5671	0.1034	2.6705	######	0.0000		0.0000			19.7777778	0.3100000	2.9300000	4.6625222	19.777778	0.310000	2.930000	4.662522
##########	CL2	23	900	60.00	0.00	3.4855	0.1295	3.6150	######	40.0000		5.6000	########	########	16.5000000	0.3800000	3.0400000	2.4422735	34.220171	0.356822	2.854575	2.293307
##########	CL2	23	900	11.00	0.00	3.8142	0.0990	3.9132	######	0.0000		0.0000			33.0555556	0.4300000	3.1300000	1.2211568	33.055556	0.430000	3.130000	1.221137
##########	CL2	23	900	37.00	0.00	3.9654	0.1262	4.0916	######	0.0000		0.0000	########	########	29.5000000	0.4400000	3.1500000	0.8880995	29.500000	0.440000	3.150000	0.888099
##########	CL2	23	900	19.00	0.00	4.0829	0.1567	4.2396	######	37.0000		0.0000	########	########	25.4444444	0.3900000	3.1300000	0.7770870	24.166186	0.370407	2.972757	0.738048
##########	CL2	23	900	11.00	0.00	4.1798	0.1284	4.3082	######	0.0000		0.0000			18.8333333	0.4100000	3.1500000	0.8880995	18.833333	0.410000	3.150000	0.888099
##########	CL2	23	900	11.00	0.00	4.0797	0.1698	4.2494	######	0.0000		0.0000			13.7777778	0.3700000	3.0300000	1.1101243	13.777778	0.370000	3.030000	1.110124
##########	CL2	23	900	11.00	0.00	4.0046	0.1469	4.1515	######	0.0000		7.6000			7.0555556	0.4300000	2.8700000	1.1101243	34.466403	0.425272	2.838443	1.097918

Measured Data | Feed Concentration | Polynomial Degree

- Enter process data (orange), feed volume (green), concentration before feeding (blue), and concentration after feeding (yellow, optional)
- Accept datasets from cell lines/processes that have different settings (e.g., different feeds or measurements)
- Can add columns to include more online/offline measurements (orange), feeds (green), or chemical species (blue and yellow)

* This Excel template is different from the previous Excel batch sheet.

2

2

Excel data input (2): feed concentration and polynomial degree sheets

	A	F	G	I
1	Cell Line	CL1	CL1	CL2
2	ID	13	13	21
3	Feed Name	F01	F02	Feed 2
9	Glucose (mM)	400.0		400.0
10	Glutamine (mM)		200.0	
11	Glutamate (mM)	11.6		
15	Lactate (mM)	0.0		

Measured Data | Feed Concentration | Polynomial Degree

- Enter the composition of feeds in the "Feed Concentration" sheet
- Feed name should be consistent with the ones in the "Measured Data" sheet

	A	D	E
1	Cell Line	CL1	CL2
2	ID	13	21
3	Cell	3	3
9	Glucose	4	4
10	Glutamine	3	3
11	Glutamate	4	4
15	Lactate	5	6

Measured Data | Feed Concentration | Polynomial Degree

- Define the polynomial degree used for the polynomial regression of specific rate calculations
- Can use varying degrees for different species and cell lines/experiments (IDs)

3

3

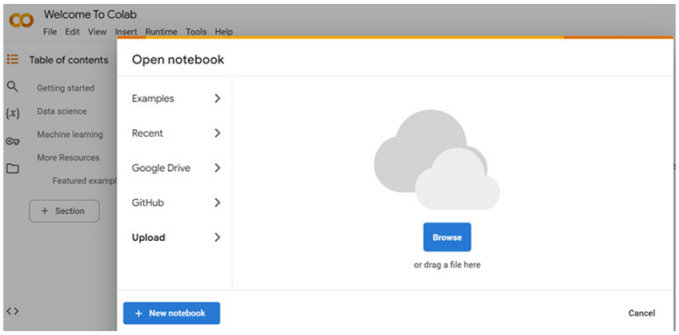
Step-by-step execution of tutorial Jupyter notebooks on Google Colab (1)

Step 1: Open Google Colab (https://colab.research.google.com/)

* Jupyter notebook is an open-source web application that allows users to create and share interactive documents that contain computational outputs

* Google Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free-of-charge access to computing resources

Step 2: Upload the "Package_Tutorial.ipynb" to Google Colab



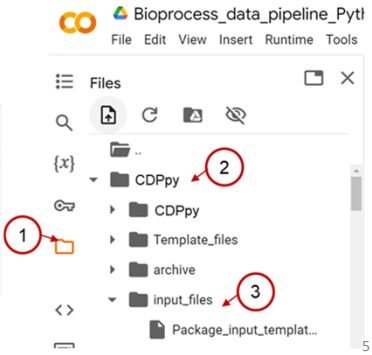
4

4

Step-by-step execution of tutorial notebooks on Google Colab (2)

Step 3: Run the first cell (grey area shown below) with the "play" bottom to setup the package

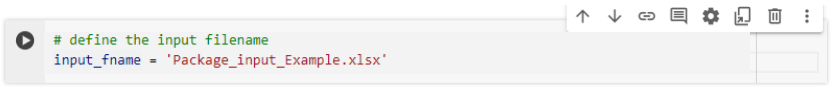
Step 4: Follow the steps in the screenshot and upload the input Excel file into the "input_files" folder



5

Step-by-step execution of tutorial notebooks on Google Colab (3)

Step 5: type the file name of the input Excel file in section 2.1



Step 6: Performing the data processing by running the rest of the notebook with "play" bottoms

- Section 2: example of processing one cell line data
- Section 3: example of processing multiple cell line data (slide 17)

Step 7: Go to section 4 to plot and analyze process profiles. (Slide 18)

6

6

Full codes for processing multi-cell-line datasets in CDApy

```
# Specify parameters that store data infomrmation for cell line 1 (CL1)
param_1 = FedBatchParameters(
          # input the cell line name, be consistent with the input excel file
          cell_line_name='CL1',
          # input "True" if there are measurements on concentrations after feeding, otherwise, "False"
          use_concentration_after_feed=False,
          # input "True" if feeding composition is known, otherwise, "False"
          use_feed_concentration=True)

# Specify parameters that store data infomrmation for cell line 2 (CL2)
param_2 = FedBatchParameters(
          # input the cell line name, be consistent with the input excel file
          cell_line_name='CL2',
          # input "True" if there are measurements on concentrations after feeding, otherwise, "False"
          use_concentration_after_feed=False,
          # input "True" if feeding composition is known, otherwise, "False"
          use_feed_concentration=True)
```

Specify data information for each cell line

```
### Repeat section 2.2
# create a Python object to preform the data processing of CL1/2
cell_line_1_2= FedBatchCellCulture()

# define path to the input data file. Change the filename if needed.
path = input_path(input_fname)

# load the data set to the data-processing object
cell_line_1_2.load_data(file=path)

# peform data processsing with specified parameters that store data infomrmation.
# pass different data information with separate input params
cell_line_1_2.perform_data_process(parameters=[param_1, param_2])
```

Input and process data

```
# export the processed dataset
cell_line_1_2.save_excel(file_name='output_CL1_2.xlsx')
```

Save results to Excel

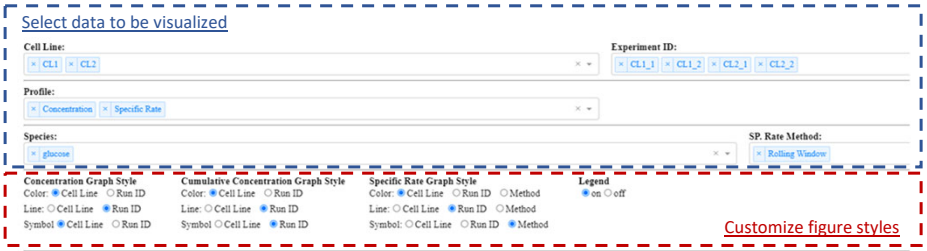
* More details can be found in sections 2 & 3 in tutorial Jupyter notebook

7

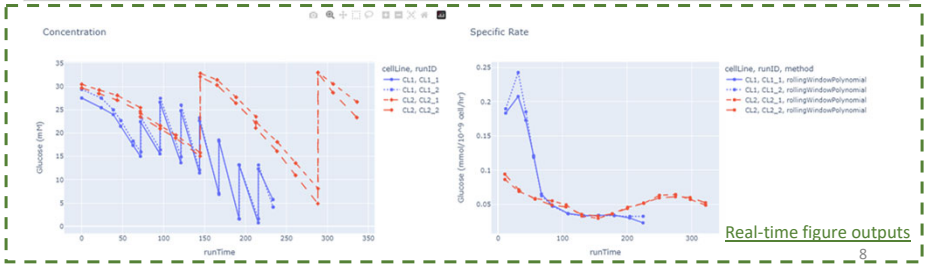
7

Interactive Plotting Widget – Customized Data Visualization

Select data to be visualized



Customize figure styles



Real-time figure outputs

8

* Result using a different dataset from the example one

8