

GROWTH CURVE ANALYSIS PIPELINE


▼ Select the file you want to analyze

```
#Import the required libraries
import os
import fnmatch
import pandas as pd

#List all csv files in the directory. The path to the directory should be specific
#and to the location where you stored your files of interest
path_to_data = "/home/nzarevsk/Desktop/iGEM/Data Analysis/EpiGrow/fwdo_ddata/"

list_of_files = os.listdir(path_to_data)
csv_files = fnmatch.filter(list_of_files, '*.csv')

for file in csv_files:
    print(file)

 Exp_Temp_35C_Growth_Sepi_20200921_preprocessed.csv
Exp_Temp_38C_Growth_Sepi_20200928_preprocessed.csv
Exp_pH_all_Growth_Sepi_20200929_preprocessed.csv
Exp_Temp_32C_Growth_Sepi_20200915_preprocessed.csv
Exp_Temp_37C_Growth_Sepi_20200930_preprocessed.csv
Exp_Temp_33C_Growth_Sepi_20200919_preprocessed.csv
Exp_Temp_34C_Growth_Sepi_20200920_preprocessed.csv
Exp_Temp_36C_Growth_Sepi_20200922_preprocessed.csv

#Choose the filename, from the list above, to be analyzed
filename = "Exp_pH_all_Growth_Sepi_20200929_preprocessed.csv"

data = pd.read_csv(path_to_data+filename)
print(data)
```

Cycle Nr.	1	2.0	3.0	4	5	
0	Time [s]	0.0000	3630.9000	7261.9000	10892.8000	14524.6000
1	Temp. [°C]	37.4000	37.1000	37.3000	37.1000	37.2000
2	A2	NaN	NaN	NaN	NaN	NaN
3	A3	NaN	NaN	NaN	NaN	NaN
4	A4	NaN	NaN	NaN	NaN	NaN
5	A5	NaN	NaN	NaN	NaN	NaN
6	BLANK_1	0.1168	0.1257	0.1249	0.1275	0.1291
7	PH4_1	0.1032	0.1041	0.1053	0.1080	0.1090
8	PH6_1	0.1013	0.1135	0.1727	0.2509	0.3838
9	PH8_1	0.0982	0.1218	0.2053	0.4240	0.6471
10	BLANK_2	0.0948	0.0980	0.1018	0.1007	0.0994
11	PH4_2	0.1016	0.1042	0.1065	0.1051	0.1069
12	PH6_2	0.1003	0.1112	0.1470	0.2395	0.3790
13	PH8_2	0.1079	0.1204	0.2057	0.4273	0.6386
14	BLANK_3	0.0948	0.0950	0.0956	0.0959	0.0957
15	PH4_3	0.1054	0.1068	0.1103	0.1049	0.1078
16	PH6_3	0.1011	0.1083	0.1489	0.2507	0.4059
17	PH8_3	0.1027	0.1194	0.2041	0.4149	0.6175
18	PH3_1	0.1013	0.1027	0.1035	0.1020	0.1009
19	PH5_1	0.0987	0.0992	0.1048	0.1099	0.1174
20	PH7_1	0.0996	0.1110	0.1684	0.3399	0.4957
21	BLANK_4	0.0949	0.0937	0.0953	0.0943	0.0941
22	PH3_2	0.1049	0.1038	0.1028	0.1078	0.1028
23	PH5_2	0.0988	0.1028	0.1073	0.1148	0.1321
24	PH7_2	0.0998	0.1101	0.1728	0.3188	0.5048
25	BLANK_5	0.0933	0.0950	0.0930	0.0972	0.0976
26	PH3_3	0.1067	0.1078	0.1064	0.1079	0.1080
27	PH5_3	0.0985	0.1019	0.1033	0.1145	0.1232
28	PH7_3	0.0993	0.1103	0.1629	0.3033	0.5305
29	BLANK_6	0.0935	0.0953	0.0937	0.0938	0.0938
30	H2	NaN	NaN	NaN	NaN	NaN
31	H3	NaN	NaN	NaN	NaN	NaN
32	H4	NaN	NaN	NaN	NaN	NaN
33	H5	NaN	NaN	NaN	NaN	NaN
34	End Time:	NaN	NaN	NaN	NaN	NaN

	6.0	7.0	8	9.0	...	14.0
0	18155.6000	21786.5000	25417.4000	29048.3000	...	47202.8000
1	37.1000	37.1000	37.0000	36.9000	...	37.1000
2	NaN	NaN	NaN	NaN	...	NaN
3	NaN	NaN	NaN	NaN	...	NaN
4	NaN	NaN	NaN	NaN	...	NaN
5	NaN	NaN	NaN	NaN	...	NaN
6	0.1263	0.1279	0.1278	0.1297	...	0.3883
7	0.1092	0.1089	0.1092	0.1090	...	0.1041
8	0.5321	0.6485	0.7746	0.8279	...	1.2632
9	0.7105	0.8641	0.9401	1.0880	...	1.1739
10	0.1010	0.1004	0.0988	0.0984	...	0.3652
11	0.1063	0.1076	0.1058	0.1079	...	0.1081
12	0.5538	0.6853	0.8495	0.9164	...	1.2868
13	0.7620	0.8668	0.9676	1.0841	...	1.2243
14	0.0959	0.0955	0.0952	0.0956	...	0.5002
15	0.1055	0.1060	0.1061	0.1085	...	0.1098
16	0.5405	0.7025	0.7626	0.8707	...	1.3711
17	0.7472	0.8853	0.9708	1.0880	...	1.2762
18	0.1051	0.1042	0.1045	0.1052	...	0.1049
19	0.1304	0.1561	0.1740	0.1780	...	0.4048
20	0.6660	0.7044	0.8467	0.9710	...	1.3335
21	0.0967	0.0960	0.1044	0.1306	...	1.0213
22	0.1063	0.1046	0.1036	0.1040	...	0.1038

23	0.1354	0.1576	0.1745	0.2001	...	0.3069
24	0.6524	0.7408	0.8302	0.9060	...	1.2580
25	0.0969	0.1020	0.1075	0.1162	...	1.1438
26	0.1127	0.1029	0.1050	0.1117	...	0.1021
27	0.1403	0.1529	0.1540	0.1964	...	0.3211
28	0.6798	0.7693	0.8706	0.9579	...	1.2100
29	0.0941	0.0961	0.0966	0.1058	...	1.0252
30	NaN	NaN	NaN	NaN	...	NaN
31	NaN	NaN	NaN	NaN	...	NaN
32	NaN	NaN	NaN	NaN	...	NaN
33	NaN	NaN	NaN	NaN	...	NaN
34	NaN	NaN	NaN	NaN	...	NaN

	15.0	16.0	17.0	18.0	19.0	20.0	21.0	\
0	50834.6000	54465.5000	58096.4000	61727.3	65358.2	68989.1	72620.0	
1	37.1000	37.2000	36.9000	37.2	36.7	37.1	37.1	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
6	0.6063	0.7102	0.7866	NaN	NaN	NaN	NaN	
7	0.1041	0.1033	0.1029	NaN	NaN	NaN	NaN	
8	1.2571	1.2457	1.3026	NaN	NaN	NaN	NaN	
9	1.2434	1.1947	1.1844	NaN	NaN	NaN	NaN	
10	0.5828	0.6957	0.8302	NaN	NaN	NaN	NaN	
11	0.1043	0.1041	0.1115	NaN	NaN	NaN	NaN	
12	1.4351	1.3989	1.4390	NaN	NaN	NaN	NaN	
13	1.2092	1.1560	1.1628	NaN	NaN	NaN	NaN	
14	0.7493	0.8894	1.0460	NaN	NaN	NaN	NaN	
15	0.1088	0.1004	0.1090	NaN	NaN	NaN	NaN	
16	1.4433	1.4742	1.3939	NaN	NaN	NaN	NaN	
17	1.2822	1.2782	1.2633	NaN	NaN	NaN	NaN	
18	0.1011	0.1002	0.1019	NaN	NaN	NaN	NaN	
19	0.4342	0.4474	0.4760	NaN	NaN	NaN	NaN	
20	1.2782	1.1865	1.2141	NaN	NaN	NaN	NaN	
21	1.0698	1.1694	1.2351	NaN	NaN	NaN	NaN	
22	0.1072	0.1048	0.1036	NaN	NaN	NaN	NaN	
23	0.3413	0.3583	0.5002	NaN	NaN	NaN	NaN	
24	1.2062	1.2838	1.2469	NaN	NaN	NaN	NaN	
25	1.2907	1.3680	1.4354	NaN	NaN	NaN	NaN	
26	0.1032	0.1049	0.1038	NaN	NaN	NaN	NaN	
27	0.3778	0.3730	0.4606	NaN	NaN	NaN	NaN	
28	1.2427	1.3121	1.1986	NaN	NaN	NaN	NaN	
29	1.0732	1.2472	1.3234	NaN	NaN	NaN	NaN	
30	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
32	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
33	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
34	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	22.0	23.0
0	76251.0	79881.9
1	36.9	37.1
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN
5	NaN	NaN
6	NaN	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN
10	NaN	NaN

```

11      NaN      NaN
12      NaN      NaN
13      NaN      NaN
14      NaN      NaN
15      NaN      NaN
16      NaN      NaN
17      NaN      NaN
18      NaN      NaN
19      NaN      NaN
20      NaN      NaN
21      NaN      NaN

```

▼ Looking at the blanks

```

26      NaN      NaN

```

```

#Import the required libraries
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline

#Retrieve the time span of the data points
Time_points = pd.to_numeric(data.loc[0][1:].values)/3600

#Retrieve the OD of the blanks
Blank_wells = ['BLANK_4', 'BLANK_5', 'BLANK_6']

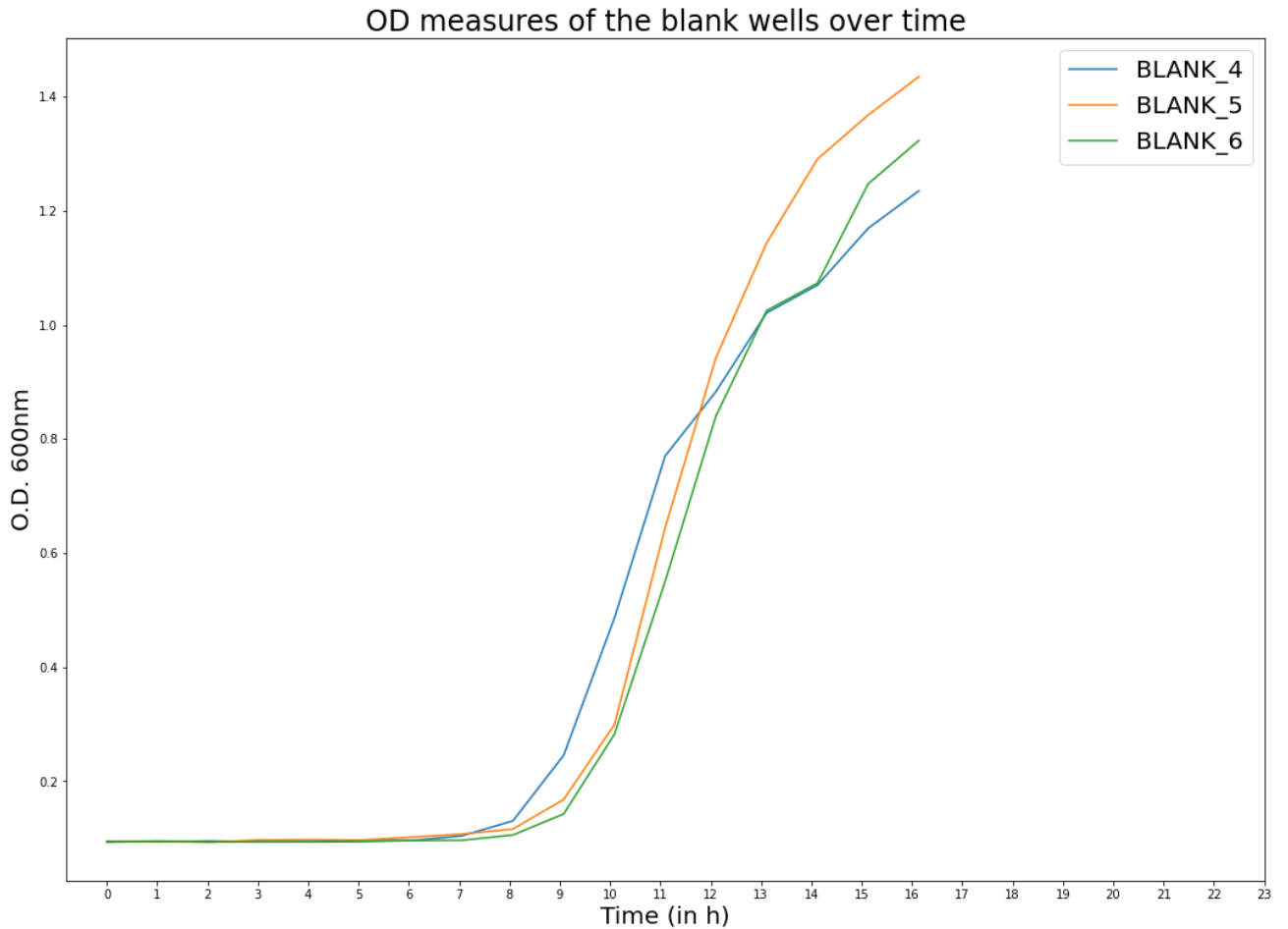
OD_blanks = data[data['Cycle Nr.'].isin(Blank_wells)]

fig,ax = plt.subplots(figsize=(18,13))

for well in OD_blanks.index:
    OD_values_blanks = pd.to_numeric(OD_blanks.loc[well][1:].values)
    plt.plot(Time_points, OD_values_blanks)

plt.xticks(np.arange(0, len(Time_points)+1))
plt.xlabel('Time (in h)', size=20)
plt.ylabel('O.D. 600nm', size=20)
plt.title('OD measures of the blank wells over time', size = 24)
plt.legend(labels=Blank_wells, fontsize=20)
plt.show()

```



#If you want to save the figure, run the following cell:

```
path_to_save = "/home/nzarevsk/Desktop/iGEM/Data Analysis/EpiGrow/"
name_of_plot = 'Plot of Blank wells'+filename.split('_preprocessed')[0]+'.png'
fig.savefig(path_to_save+name_of_plot, bbox_inches='tight')
```

#Look at the mean OD of the blank wells

```
OD_blanks_mean = OD_blanks.drop(labels='Cycle Nr.', axis = 1)
OD_blanks_mean = OD_blanks_mean.astype(float)
```

```
fig,ax = plt.subplots(figsize=(18,13))
```

```
for well in OD_blanks.index:
```

```
    OD_values_blanks = pd.to_numeric(OD_blanks.loc[well][1:].values)
    plt.plot(Time_points, OD_values_blanks, alpha=0.5)
```

```
plt.plot(Time_points, OD_blanks_mean.mean(), c='r', alpha=1, lw = 2)
```

```
plt.xticks(np.arange(0,len(Time_points)+1))
```

```
plt.xlabel('Time (in h)', size=20)
```

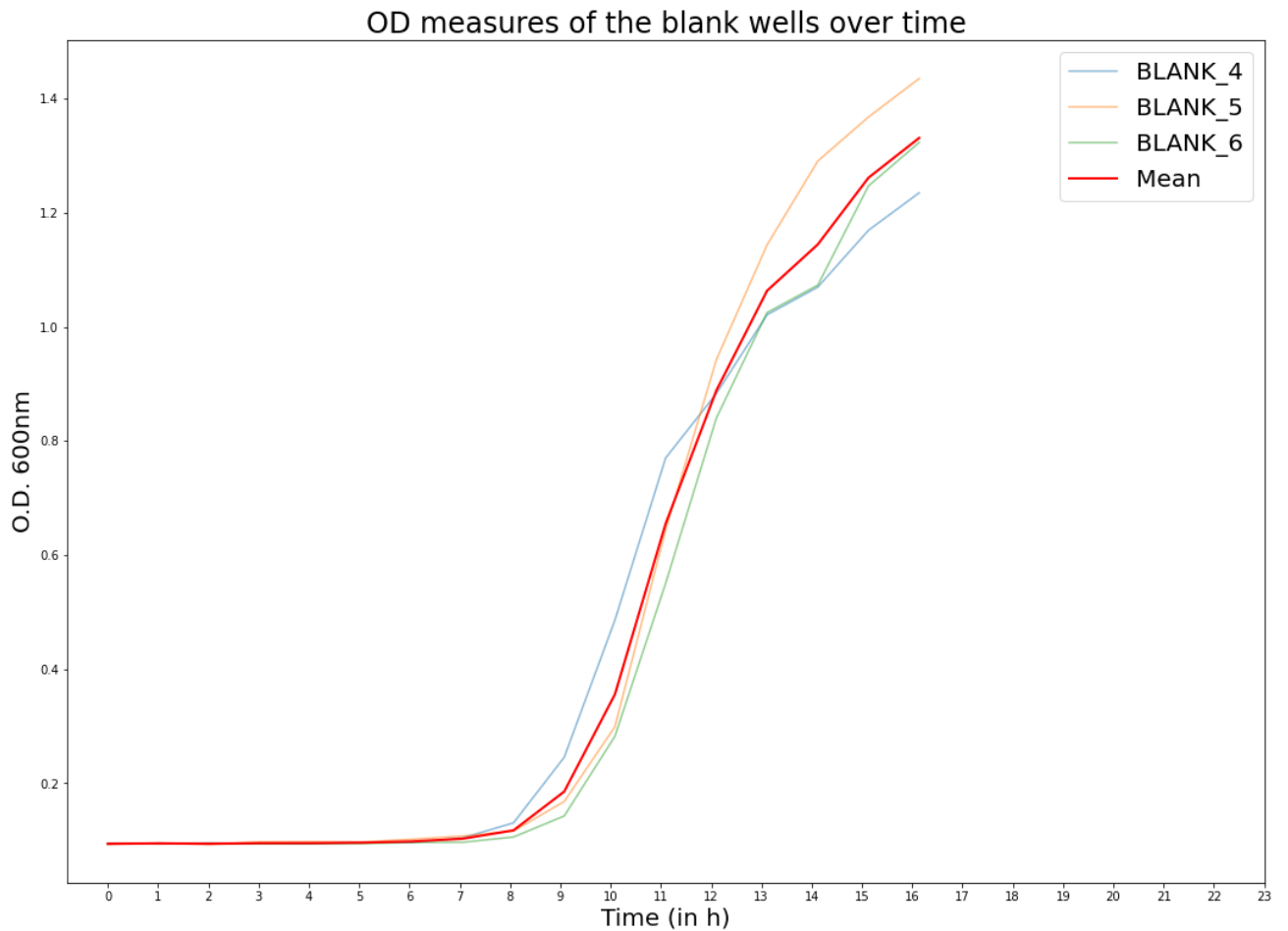
```
plt.ylabel('O.D. 600nm', size=20)
```

```
plt.title('OD measures of the blank wells over time', size = 24)
```

```
Blank_wells = np.append(Blank_wells,'Mean')
```

```
plt.legend(labels=Blank_wells, fontsize=20)
```

```
plt.show()
```



#If you want to save the figure, run the following cell:

```
path_to_save = "/home/nzarevsk/Desktop/iGEM/Data Analysis/EpiGrow/"
name_of_plot = 'Plot of Blank wells wMean_odd'+filename.split('_preprocessed')[0]+
fig.savefig(path_to_save+name_of_plot, bbox_inches='tight')
```

#Get the average Blank OD to get rid of the contamination

```
Stop_T = 7
```

```
Corrected_Average_OD_blanks = OD_blanks_mean.mean()[ :Stop_T].mean()
```

▼ Looking at the samples

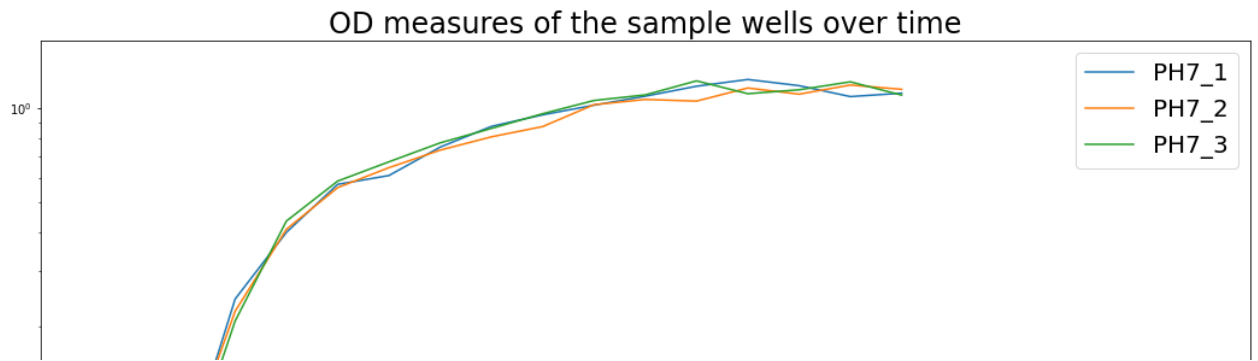
```
#Retrieve the OD of the samples
Sample_wells = ['PH7_1', 'PH7_2', 'PH7_3']

OD_samples = data[data['Cycle Nr.'].isin(Sample_wells)]

fig, ax = plt.subplots(figsize=(18, 13))

for well in OD_samples.index:
    OD_values_samples = pd.to_numeric(OD_samples.loc[well][1:].values)
    #DO NOT FORGET TO SUBSTRACT THE OD FROM THE BLANKS AND PLOT ON A SEMILOG PLOT
    plt.semilogy(Time_points, (OD_values_samples - [Corrected_Average_OD_blanks]) * len

plt.xticks(np.arange(0, len(Time_points) + 1))
plt.xlabel('Time (in h)', size=20)
plt.ylabel('O.D. 600nm', size=20)
plt.title('OD measures of the sample wells over time', size = 24)
plt.legend(labels=Sample_wells, fontsize=20)
plt.show()
```



#If you want to save the figure, run the following cell:

```
path_to_save = "/home/nzarevsk/Desktop/iGEM/Data Analysis/EpiGrow/"
name_of_plot = 'Growth Curves pH7_'+filename.split('_preprocessed')[0]+'.png'
fig.savefig(path_to_save+name_of_plot, bbox_inches='tight')
```

#Look at the mean OD of the sample wells

```
OD_samples_mean = OD_samples.drop(labels='Cycle Nr.', axis = 1)
OD_samples_mean = OD_samples_mean.astype(float)
```

```
fig,ax = plt.subplots(figsize=(18,13))
```

```
for well in OD_samples.index:
```

```
    OD_values_samples = pd.to_numeric(OD_samples.loc[well][1:].values)
```

```
    #DO NOT FORGET TO SUBSTRACT THE OD FROM THE BLANKS AND PLOT ON A SEMILOG PLOT
    plt.semilogy(Time_points, (OD_values_samples-[Corrected_Average_OD_blanks]*len
```

```
Mean_OD_samples_corrected = OD_samples_mean.mean()-[Corrected_Average_OD_blanks]*len
```

```
plt.semilogy(Time_points, Mean_OD_samples_corrected, alpha=1, c='r', lw=2)
```

```
plt.xticks(np.arange(0,len(Time_points)+1))
```

```
plt.xlabel('Time (in h)', size=20)
```

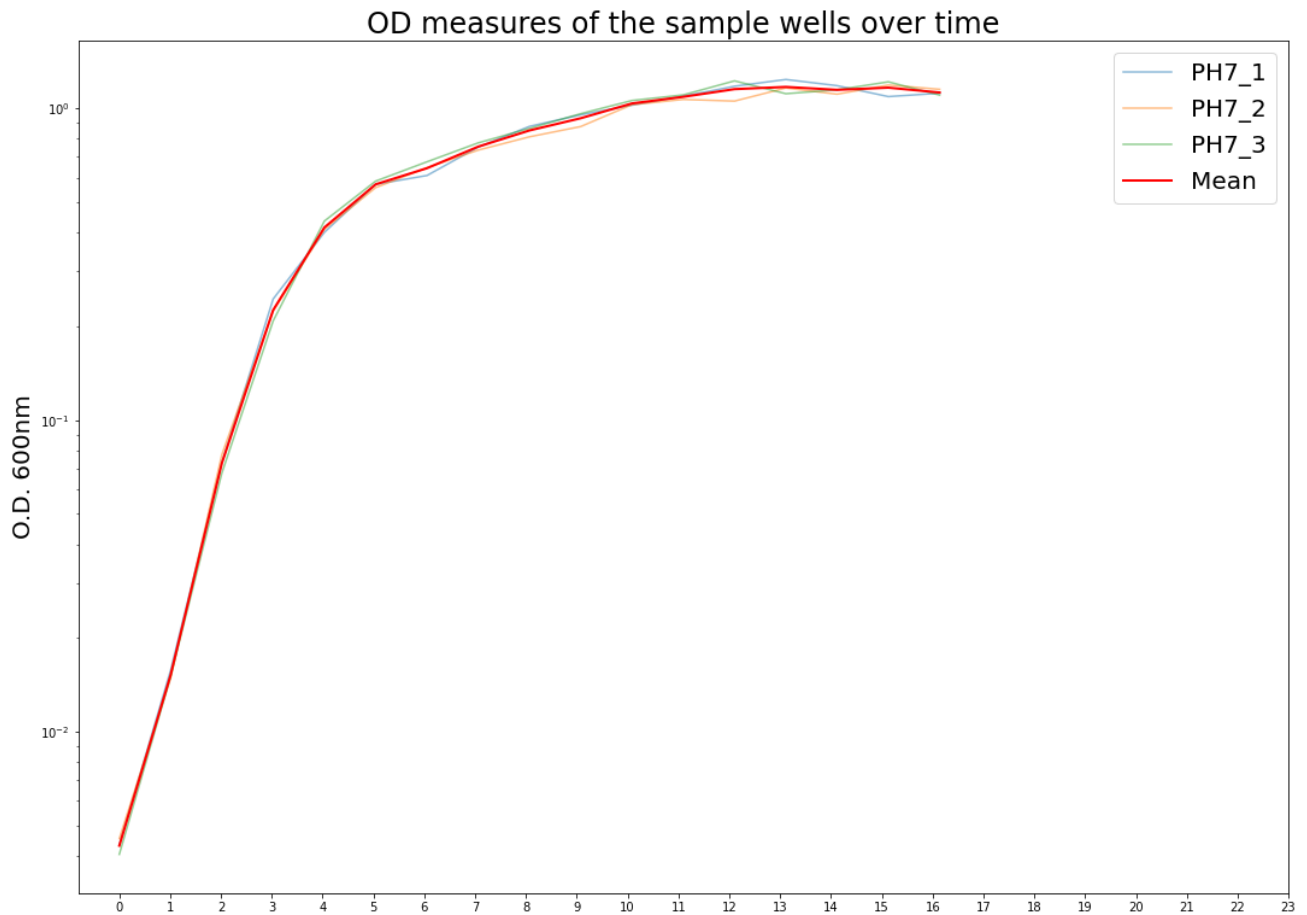
```
plt.ylabel('O.D. 600nm', size=20)
```

```
plt.title('OD measures of the sample wells over time', size = 24)
```

```
Sample_wells = np.append(Sample_wells,'Mean')
```

```
plt.legend(labels=Sample_wells, fontsize=20)
```

```
plt.show()
```

#If you want to save the figure, run the following cell:

```
path_to_save = "/home/nzarevsk/Desktop/iGEM/Data Analysis/EpiGrow/"
name_of_plot = 'Growth Curves pH7 wMean_'+filename.split('_preprocessed')[0]+'.png'
fig.savefig(path_to_save+name_of_plot, bbox_inches='tight')
```

▼ Calculate the Growth Rate and the Generation Time

#To extract the Growth Rate (noted GR), we will need to use the graphical informat:
#We will need to select the frame of the so-called exponential phase of the growth

```
#To obtain the Growth Rate r, we will use the following formula  $dN/dt = r*N$ 
# -->  $dN/N = r*dt$ 
# -->  $\log(N) - \log(N_0) = r*t - r*t_0$ 
# -->  $\log(N) - \log(N_0) = r*(t-t_0)$ 
# -->  $r = (\log(N) - \log(N_0)) / (t-t_0)$ 
```

#Select the time points that define the exponential phase frame:

```
Exp_phase_Start = 0
Exp_phase_End = 4
```

```
Exp_phase_value_start = Mean_OD_samples_corrected[Exp_phase_Start]
Exp_phase_value_end = Mean_OD_samples_corrected[Exp_phase_End]
```

```
Time_start = Time_points[Exp_phase_Start]
Time_end = Time_points[Exp_phase_End]
```