

Research Project

CEE 4530 Group 9 Erin Holbrook (eh558): 5 hours & Yeonsoo Park (yp273): 5 hours

Introduction

Adsorption is a process in which surface-active materials in true solution are removed from the solvent by interphase transfer to the surface of an adsorbent particle [1]. The process depends on various factors such as surface area, pore size distribution and surface functional groups of the adsorbent; polarity, solubility and molecule size of adsorbate; solution pH and presence of other ions in solution and so on [2]. Among those factors, some studies have looked into the effect of the molecular weight of the adsorbent [3]. Also, a number of studies researched competitive adsorption onto different adsorbates [4].

For the adsorption lab, we plotted the adsorption curve of Red #40 to activated carbon and sand as a function of time. The main purpose of the lab was to understand how long it would take for the activated carbon to fail to filter the influent dye by plotting the adsorption curve. Though this lab was meaningful in that it provided insight into the efficiency of activated carbon to filter a certain substance, it is far from being a model of the real world.

Activated carbon is a commonly used adsorbent in filtering systems ranging from drinking water treatment plants to household water filters and even air purifiers. In all such systems, activated carbon is used not to treat merely a single contaminant but multiple at once. Therefore, it is most likely that the contaminants interfere with each other when being adsorbed to the adsorbent. To better understand the adsorption under more realistic conditions, this project studied the competitive adsorption of different dyes onto an activated carbon column.

Objectives

The first objective was to verify that there is an observable difference in the adsorption curve when dyes of similar and different molecular weights are filtered through the filter. We further anticipate that with the increase of molecular weight, less dye particles will adsorb and breakthrough faster. The second objective was to observe the effect of competition on the adsorption onto the activated carbon by doubling the dye stock's concentration supplied to the filter.

Experiment

Key Design Parameters

Four different dyes were selected based on the absorbance peak and molecular weight of each dye. The absorbance peak was considered because of the possibility of installing more than one photometer. Had we used multiple photometers, it would have been crucial that the dyes did not have an overlapping absorbance peak. As summarized in Table 1, the molecular weights of Methylene Blue, Carminic Acid, and Erythrosine B are less than, similar to, and greater than that of R40, which was considered to be the base case when designing our experiment.

Dye	Absorbance Peak (nm)	Molecular Weight (g/mol)
Methylene Blue	665	319.85
Carminic Acid (R4)	293	492.38
Allura Red (R40)	485	496.42
Erythrosine (R3)	354	879.86

Table 1: Absorbance peak and molecular weight of each dye

To investigate the effect of adding more competition among the same dye particles entering the filter column, we experimented the four dyes at two different concentrations, 50 and 100 mg/L.

For all experiments, the pump's speed and filter composition were set to be constant. The pump was set to 10 rpm and the filter consisted of 5g of activated carbon and 105 g of sand. To ensure that the filter materials are well mixed, we stirred up the activated carbon and sand together before pouring it into the filter column.

Setup

1. Begin by setting up the carbon column and feed system as shown in the following image. This setup is the same as the one used in the Adsorption lab.

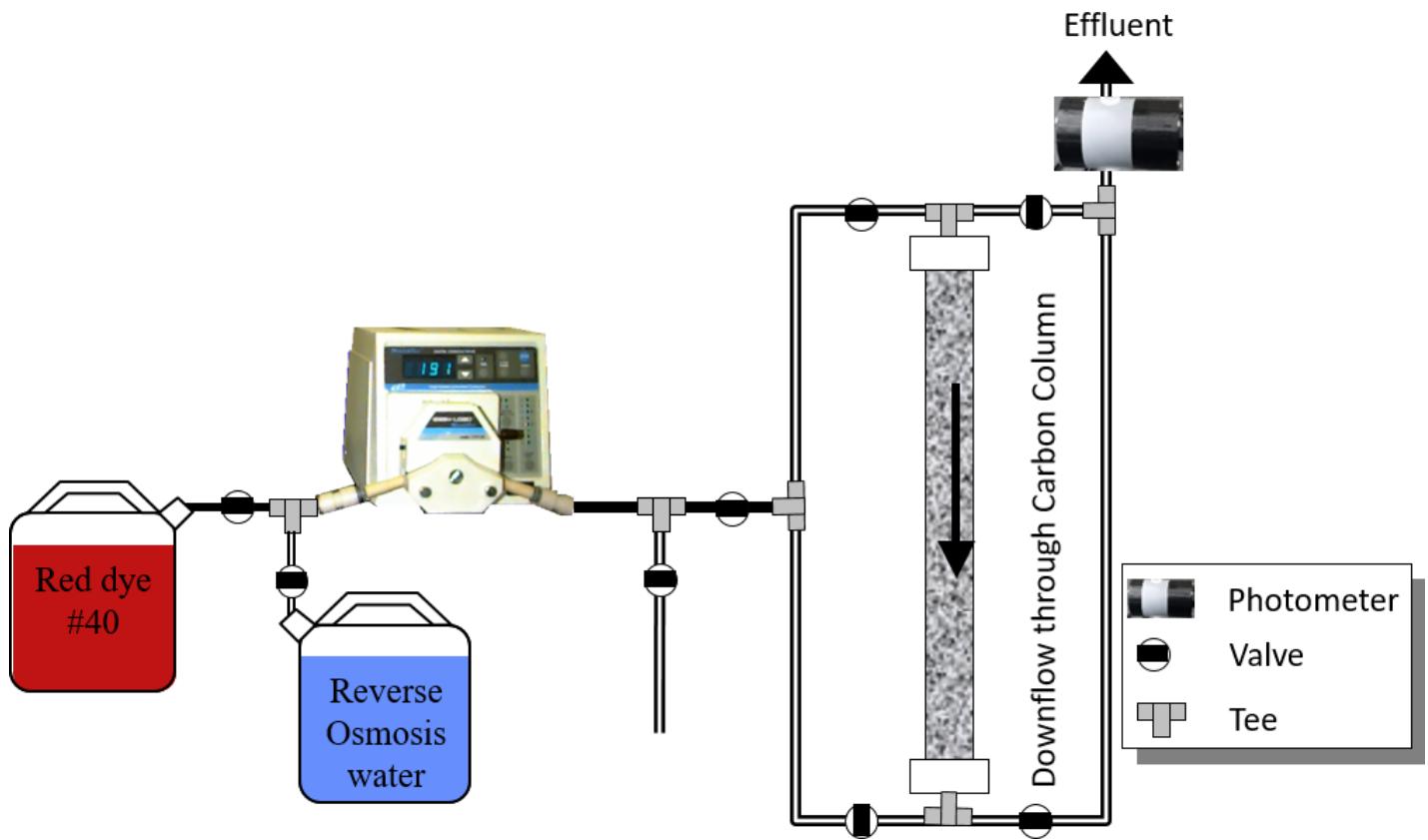


Figure 1: Experiment Schematic [1]

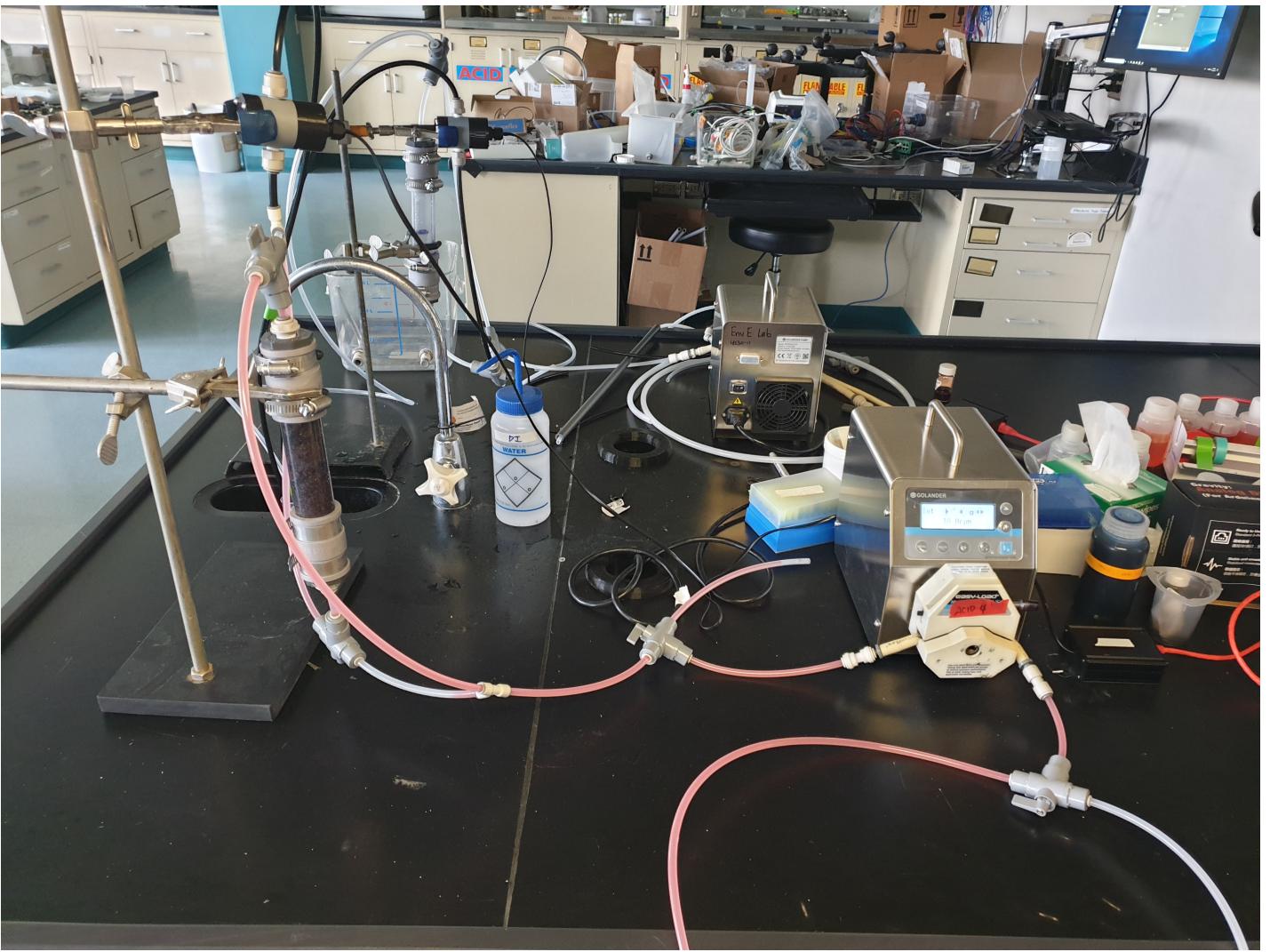


Figure 2: Experiment Setup

2. For first experiment, use R40 at 50 mg/L concentration for dye stock. Use 5 grams of activated carbon mixed well with 105 grams of sand in filter.
3. Measure the concentration over time of the dye using a photometer attached to the apparatus after filtering. Data will be recorded using ProCoDA, which will be set to measure at 5 second intervals.
4. Run until $C/C_o > 0.6$, then save data for analysis.
5. Repeat at 50mg/L concentration for all dyes (R3, R4, and methylene blue)
6. Repeat experiment using 100 mg/L concentration for all dyes.

Challenges

Originally, a second photometer measuring at a different wavelength was expected to be available to enable us preparing a mixed stock of dyes. However, because we could not attain such photometer and there was no other way to measure mixed dyes separately, we decided to modify the original plan and double each dyes concentration instead of mixing the dyes to ensure.

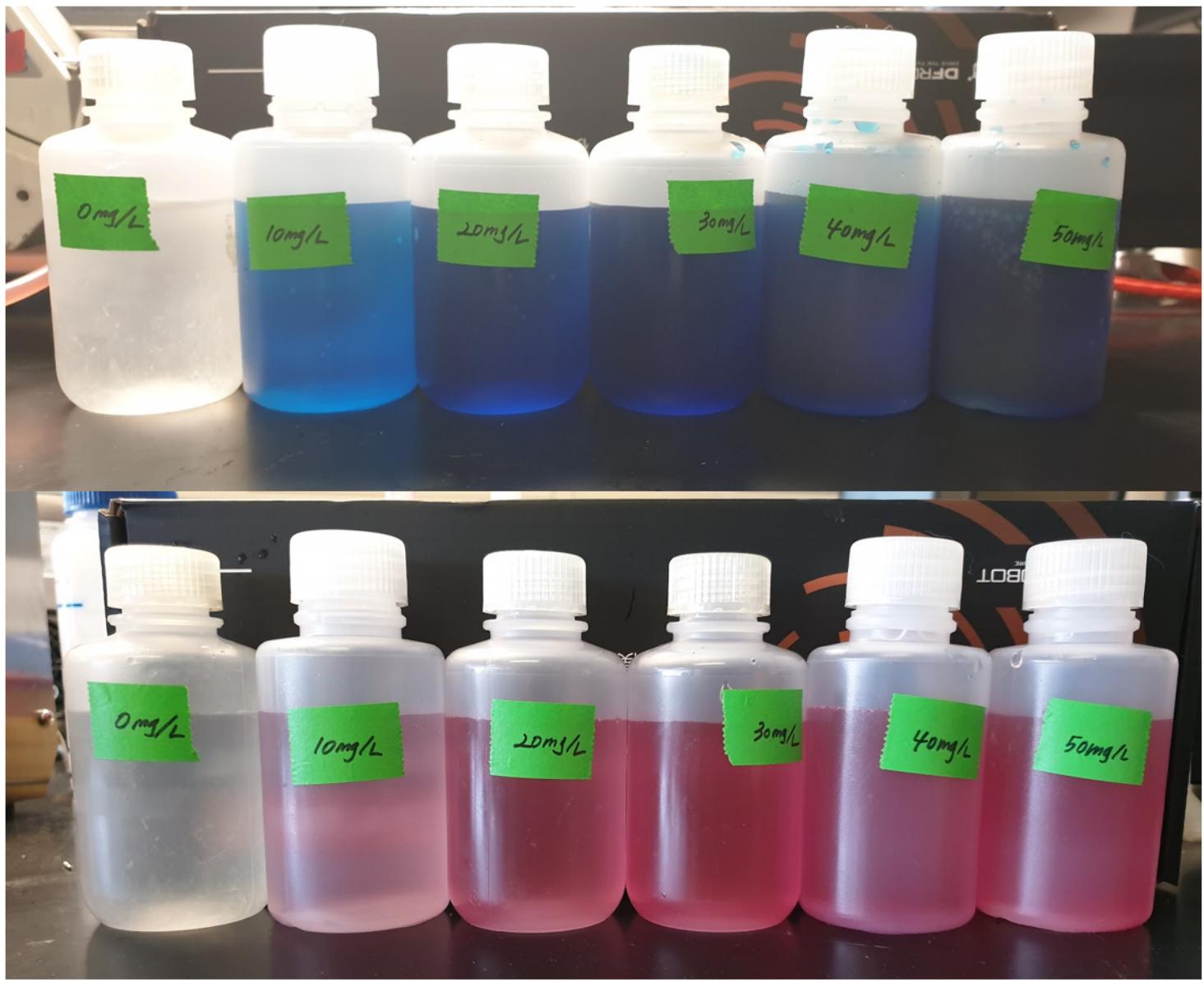


Figure 3: Calibration standards of Methylene Blue (top) and Etyrosin B (bottom)

Only half way through the experiments did we realize the change's effect on calibrating the photometer. Because we originally planned that each dye would only have a concentration of 50 mg/L, we prepared standards of each dye only up to 50 mg/L. However, with the change, we were working with 100 mg/L of each dye as well, which we did not directly calibrate for. Still, because all four calibration curves had a linear trend and R-squared values greater than 0.995, given the limited time available, we decided to proceed with the calibration curve that we had for each dye instead of extending them and repeating all experiments.

Moreover, we observed suspended precipitants in the carminic acid stock which may have affected the photometer's readings, and thus did not proceed into the 100 mg/L experiment for carminic acid.

Results

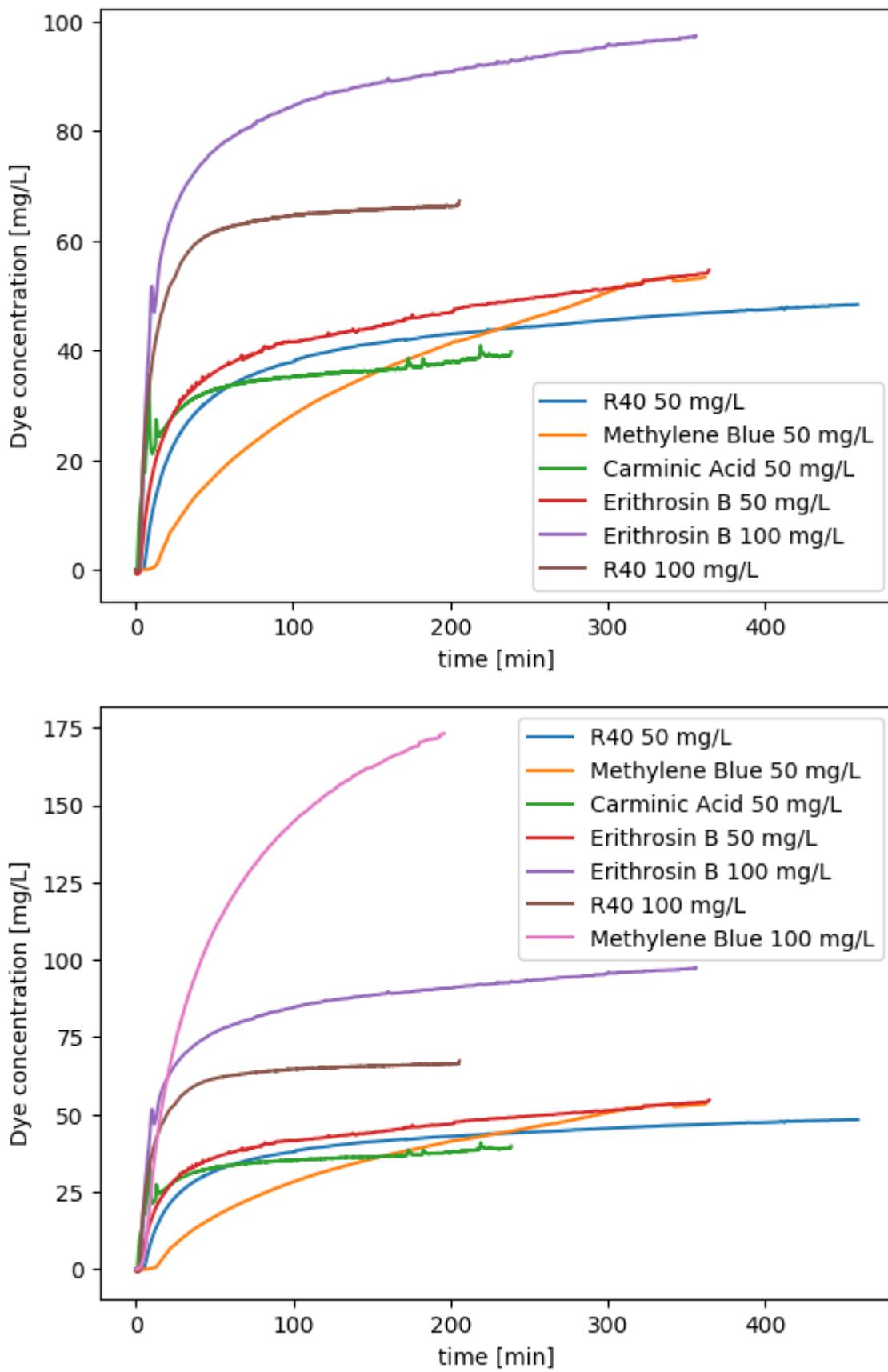


Figure 4: Breakthrough curves of successful experiments (top) and all experiments (bottom)

The breakthrough curves in Figure 4 show that all experiments concaved and plateaued at a certain concentration. Only six of the eight cases experimented were successful. Carminic Acid at 50 mg/L

showed cloudiness. Therefore, we did not have certainty of the measurement and did not proceed to the experiment with the dye at a higher concentration. For the Methylene B 100 mg/L case, as shown in Figure 5, the dye concentration plateaued at a value much higher than the stock concentration. A possible explanation would be that we were measuring a concentration out of the calibration range. Another explanation is that there could have been an error in preparing the stock.

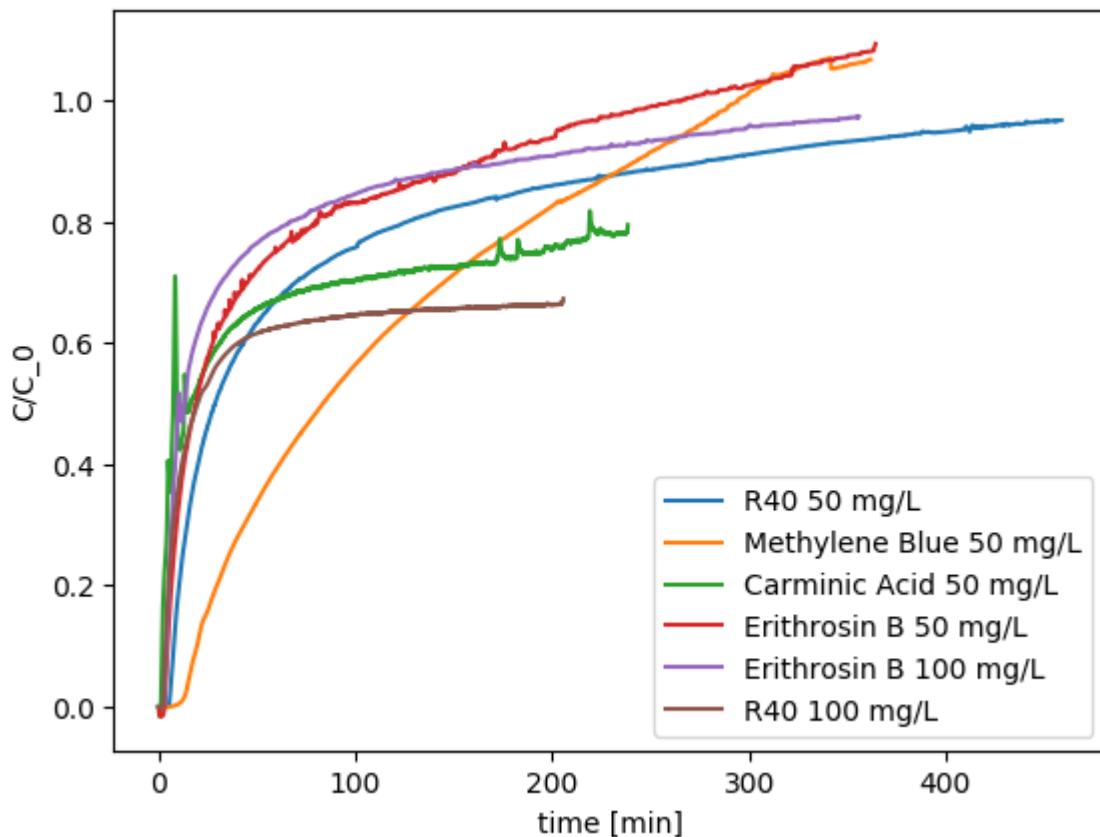


Figure 5: C/C_0 versus t/θ

Still, as shown in figure 5, we were able to see that all successful experiments reached a C/C_0 ratio greater than 0.6 and even 1 for some cases.

Table 1: Adsorption parameters for each experiment

Experiment	$R_{adsorption}$	q_0	breakthrough time
R40 50 mg/L	6.543	0.0017	26.017
Methylene Blue 50 mg/L	19.059	0.0054	75.782
Carminic Acid 50 mg/L	1.674	0.0002	6.655
Erythrosine B 50 mg/L	4.299	0.0010	17.093
Erythrosine B 100 mg/L	2.225	0.0004	8.849

Experiment	$R_{adsorption}$	q_0	breakthrough time
Methylene Blue 100 mg/L	3.614	0.0016	14.37
R40 100 mg/L	4.508	0.0011	17.924

The table above summarizes the adsorption parameters computed for each experiment. All three parameter values were highest for Methylene Blue, followed by R40, Erythrosine B, and Carminic Acid. At first glance, such ranking may not seem to have any correlation with the molecular weights of these dyes. However considering that the result for carminic acid is not reliable, we can indeed observe that as the molecular weight increases, all three parameter values decrease, as shown in Figure 6. The decrease in the parameters' values indicates that less adsorption occurs and therefore the breakthrough happens much earlier than other cases.

Comparing the 50 mg/L and 100 mg/L experiments, we were able to observe that the breakthrough time decreased with the increase of concentration as expected. However, the decrease was not 50 % despite the doubled stock concentration. The breakthrough times decreased by 81.04 %, 31.11 % and 48.23 % for Methylene Blue, R40, and Erythrosine B, respectively.

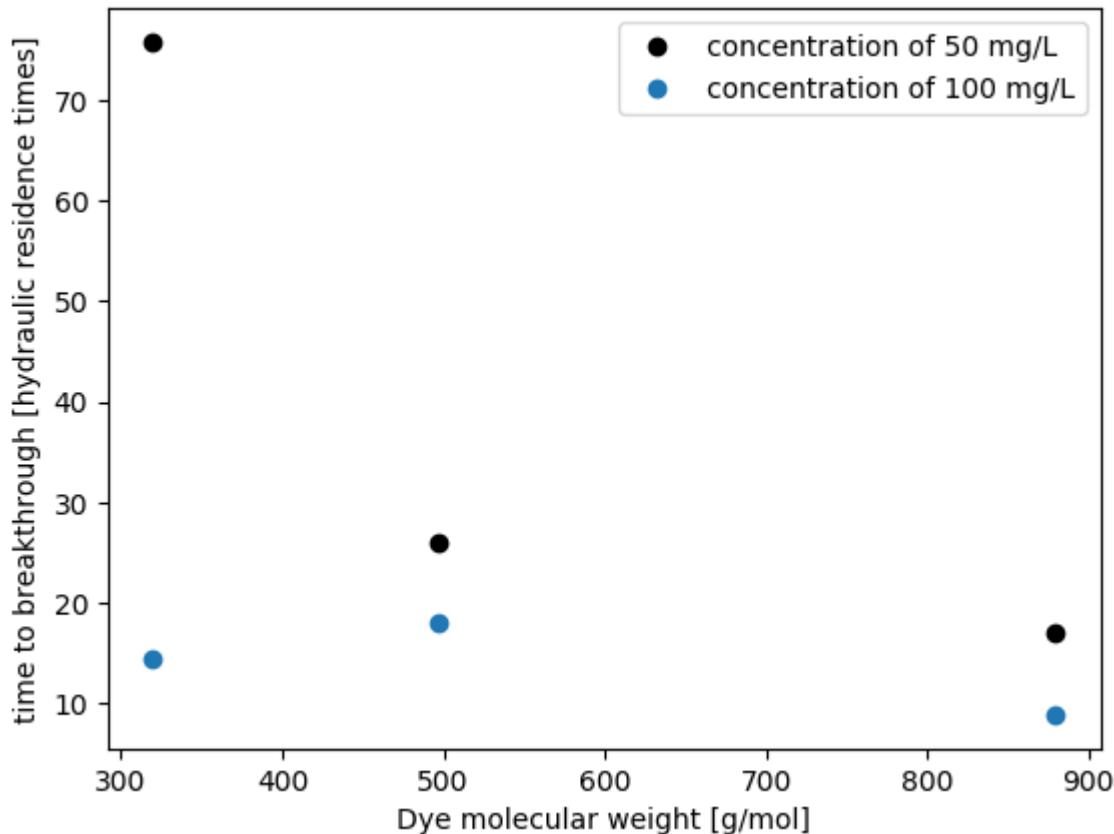


Figure 6: Breakthrough time versus molecular weight

Conclusion

From the experiments, we observed that the adsorption does change as a function molecular weight. Indeed as the molecular weight increased, the breakthrough occurred much sooner. Moreover, when increasing the competition of each dye by increasing the stock concentration, we were able to observe a decrease in breakthrough time.

Discussion

Though the experiments did give us insight into how molecular weight may affect the adsorption curves, due to the cloudiness of the carminic acid experiment, we were not able to confirm that when the molecular weight is similar, the adsorption curves would be similar as well. Therefore, we could not conclude that the molecular weight is the dominant factor in adsorption to activated carbon.

Moreover, because the calibration curves range up to 50 mg/L, it would be meaningful to repeat the higher concentration experiments with a corrected calibration curve.

Finally, because our original goal was to measure competition between different adsorbates, a more realistic study on competitions effect on adsorption would require experiments with mixed dye stocks and devices that can differentiate and measure the different dyes, individually.

References

1. CEE 4530 Textbook
2. Radovic LR, Moreno-Castilla C, Rivera-Utrilla J. Carbon materials as adsorbents in aqueous solutions. In: Radovic LR, editor. Chemistry and physics of carbon, vol. 27. Marcel Dekker; 2001. p. 228 - 405.
3. Davis, James A., and Rolf Gloor. "Adsorption of dissolved organics in lake water by aluminum oxide. Effect of molecular weight." Environmental science & technology 15.10 (1981): 1223-1229.
4. Newcombe, G., et al. "Simultaneous adsorption of MIB and NOM onto activated carbon: II. Competitive effects." Carbon 40.12 (2002): 2147-2156.
5. Morlock, Gertrud E., and Claudia Oellig. "Rapid planar chromatographic analysis of 25 water-soluble dyes used as food additives." Journal of AOAC International 92.3 (2009): 745-756.

Appendix

```

from aguaclara.core.units import unit_registry as u
import aguaclara.research.environmental_processes_analysis as epa
import aguaclara.core.physchem as pc
import aguaclara.core.utility as ut
import numpy as np
import matplotlib.pyplot as plt
import collections
import os
from pathlib import Path
import pandas as pd
import csv

# generate a function to pull relevant data from raw files
def adsorption_data(C_column, dirpath):
    """This function extracts the data from folder containing tab delimited
    files of adsorption data. The file must be the original tab delimited file.

Parameters
-----
C_column : int
    index of the column that contains the dissolved oxygen concentration
    data.
dirpath : string
    path to the directory containing aeration data you want to analyze
Returns
-----
filepaths : string list
    all file paths in the directory sorted by flow rate
time_data : numpy array list
    sorted list of numpy arrays containing the times with units of seconds
Examples
-----
"""

# return the list of files in the directory
metadata = pd.read_csv(dirpath + '/metadata.txt', delimiter='\t')
filenames = metadata['file name']
# extract the flowrates from the filenames and apply units
# sort airflows and filenames so that they are in ascending order of flow rates
filepaths = [dirpath + '/' + i for i in filenames]
# C_data is a list of numpy arrays. Thus each of the numpy data arrays can have different lengths
# cycle through all of the files and extract the column of data with oxygen concentrations
# ERROR will not read experiment 8 data. I removed experiment 8 info from metadata temporarily
C_data=[epa.column_of_data(i,epa.notes(i).last_valid_index() + 1,C_column,-1,'mg/L') for i in range(len(filenames))]
time_data=[(epa.column_of_time(i,epa.notes(i).last_valid_index() + 1,-1)).to(u.s) for i in range(len(filenames))]

adsorption_collection = collections.namedtuple('adsorption_results','metadata filenames C_data time_data')
adsorption_results = adsorption_collection(metadata, filenames, C_data, time_data)
return adsorption_results

# pull and sort data
C_column = 1 # column to be read containing concentration data
dirpath = "/Users/ErinHealy/Desktop/CEE_4530/Final_project/final_data"

```

```
metadata, filenames, C_data, time_data = adsorption_data(C_column, dirpath)

def column_of_time_csv(data_file_path, start, end=-1):
    """This function extracts the column of times from a ProCoDA data file.

Parameters
-----
data_file_path : string
    File path. If the file is in the working directory, then the file name
    is sufficient.

start : int or float
    Index of first row of data to extract from the data file

end : int or float, optional
    Index of last row of data to extract from the data
    Defaults to -1, which extracts all the data in the file

Returns
-----
numpy array
    Experimental times starting at 0 day with units of days.

Examples
-----
ftime(Reactor_data.txt, 0)

"""
if not isinstance(start, int):
    start = int(start)
if not isinstance(end, int):
    end = int(end)

df = pd.read_csv(data_file_path, delimiter=',')
start_time = pd.to_numeric(df.iloc[start, 0])*u.day
day_times = pd.to_numeric(df.iloc[start:end, 0])
time_data = np.subtract((np.array(day_times)*u.day), start_time)
return time_data

def column_of_data_csv(data_file_path, start, column, end="-1", units ""):
    """This function extracts a column of data from a ProCoDA data file.

Parameters
-----
data_file_path : string
    File path. If the file is in the working directory, then the file name
    is sufficient.

start : int
    Index of first row of data to extract from the data file

end : int, optional
```

Index of last row of data to extract from the data
Defaults to -1, which extracts all the data in the file

column : int or string
int:
 Index of the column that you want to extract. Column 0 is time.
 The first data column is column 1.
string:
 Name of the column header that you want to extract

units : string, optional
The units you want to apply to the data, e.g. 'mg/L'.
Defaults to "" which indicates no units

Returns

numpy array
 Experimental data with the units applied.

Examples

column_of_data(Reactor_data.txt, 0, 1, -1, "mg/L")
"""

```
if not isinstance(start, int):
    start = int(start)
if not isinstance(end, int):
    end = int(end)

df = pd.read_csv(data_file_path, delimiter=',')
if units == "":
    if isinstance(column, int):
        data = np.array(pd.to_numeric(df.iloc[start:end, column]))
    else:
        df[column][0:len(df)]
else:
    if isinstance(column, int):
        data = np.array(pd.to_numeric(df.iloc[start:end, column]))*u(units)
    else:
        df[column][0:len(df)]*u(units)
return data
def notes_csv(data_file_path):
    """This function extracts any experimental notes from a ProCoDA data file.
```

Parameters

data_file_path : string
File path. If the file is in the working directory, then the file name
is sufficient.

Returns

dataframe

The rows of the data file that contain text notes inserted during the experiment. Use this to identify the section of the data file that you want to extract.

Examples

```
df = pd.read_csv(data_file_path, delimiter=',')
text_row = df.iloc[0:-1, 0].str.contains('[a-z]', '[A-Z]')
text_row_index = text_row.index[text_row].tolist()
notes = df.loc[text_row_index]
return notes

ex8_path = '/Users/ErinHealy/Desktop/CEE_4530/Final_project/final_data/Experiment#8_2.us-ascii.csv'
C_data8 = column_of_data_csv(ex8_path,notes_csv(ex8_path).last_valid_index() + 1,C_column,-1,'mg/L')
time_data8 = column_of_time_csv(ex8_path,epa.notes(ex8_path).last_valid_index() + 1,-1).to(u.s)
Flow_rate = ([metadata['flow (mL/s)'][i] for i in metadata.index]) * u.mL/u.s # flow rate of exper
Flow_rate8 = 0.466*u.ml/u.s
Mass_carbon = ([metadata['carbon (g)'][i] for i in metadata.index]) * u.g # mass of carbon in cyli
Mass_carbon8 = 4.596*u.g
C_0 = ([metadata['C_0 (mg/L)'][i] for i in metadata.index]) * u.mg/u.L # initial dye concentratior
C_08 = 100*u.mg/u.L

# set contants
porosity = 0.4 # assume porosity is the same for sand and for activated carbon (for simplicity).
Column_D = 1 * u.inch # diameter of column
Column_A = pc.area_circle(Column_D) # area of column
Column_L = 15.2 * u.cm # column length
Column_V = (Column_A * Column_L).to(u.mL) # column volume
HRT = (porosity * Column_V/Flow_rate).to(u.s) # estimate the hydraulic residence for all of the co
HRT8 = (porosity * Column_V/Flow_rate8).to(u.s)
# zero the concentration data by subtracting the value of the first data point from all data point
for i in range(np.size(filenames)):
    C_data[i]=C_data[i]-C_data[i][0]
C_data8 = C_data8 -C_data8[0]
# Tubing volume can be estimated using data where column contained only sand. Because sand is iner
# can be estimated based on flowrate and time to breakthrough
Sand_flowrate = []
Time_sandbreak = []
Sand_volumes = []
for i in range(np.size(filenames)):
    if (metadata['carbon (g)'][i] == 0):
        Sand_flowrate.append(Flow_rate[i])
        for k in range(np.size(C_data[i])):
            if (C_0[i]*0.50< C_data[i][k]):
                Time_sandbreak.append(time_data[i][k])
                break
Tubing_V = 0 * u.ml
```

```

for i in range(len(Sand_flowrate)):
    Sand_volumes.append(Sand_flowrate[i]*Time_sandbreak[i])
    Tubing_V = Tubing_V + Sand_volumes[i]
Tubing_V = (Tubing_V/len(Sand_flowrate))- Column_V
Tubing_HRT = Tubing_V/Flow_rate
V_total = Column_V + Tubing_V

# Find the time when the effluent concentration was 50% of the influent concentration
M = [] # vector of carbon masses
t_mtz = [] # time vector of 50% mark
t_water = [] # Volume/Flow Rate/hydraulic residence time
for i in range(np.size(filenames)):
    if (metadata['carbon (g)'][i] != 0):
        M.append(metadata['carbon (g)'][i])
        t_water.append(V_total/Flow_rate[i]/HRT[i])
        for k in range(np.size(C_data[i])):
            if ((0.5*C_0[i])< C_data[i][k]):
                t_mtz.append(time_data[i][k]/HRT[i])
                break
M.append(4.996)
t_water.append(V_total/Flow_rate8/HRT8)
for k in range(np.size(C_data8)):
    if ((0.5*C_0[i])< C_data8[k]):
        t_mtz.append(time_data8[k]/HRT[i])
        break

#checked that the number of experiments that activated carbon was used and the number of times at
print(np.size(M))
print(np.size(t_mtz))

# generate a plot of concentration over time for each experiment
for i in range(np.size(filenames)):
    time_data[i] = time_data[i]*u.min/(60*u.sec)
time_data8 = time_data8*u.min/(60*u.sec)
for i in range(np.size(filenames)):
    if (metadata['carbon (g)'][i] != 0):
        plt.plot(time_data[i], C_data[i], '--', label = filenames[i])
# plt.plot(time_data8, C_data8, '--', label = "Experiment#8")
leg = ["R40 50 mg/L", "Methylene Blue 50 mg/L", "Carminic Acid 50 mg/L", "Eriothrosin B 50 mg/L", 'mol_weight = [496.42, 319.85, 492.38, 879.86, 879.86, 496.42, 319.85]
plt.legend(leg)
plt.xlabel(r'time [min]')
plt.ylabel(r'Dye concentration [mg/L]')
plt.savefig('/Users/ErinHealy/Desktop/CEE_4530/Final_project/conc_v_time.png', bbox_inches = 'tight')
plt.show()

# generate a plot of C/C_0 over time
relativeC_data = []
for i in range(np.size(filenames)):
    relativeC_data.append(C_data[i]/C_0[i])
    if (metadata['carbon (g)'][i] != 0):

```

```

        plt.plot(time_data[i], relativeC_data[i], '--', label = filenames[i])
plt.legend(leg)
plt.xlabel(r'time [min]')
plt.ylabel(r'C/C_0')
plt.savefig('/Users/ErinHealy/Desktop/CEE_4530/Final_project/conc_over_init.png', bbox_inches = 'tight')
plt.show()

#generate a plot of Breakthrough time v. molecular weight
t_mtz_nocarminic = t_mtz
t_mtz_nocarminic = t_mtz_nocarminic.pop(2) #remove carminic acid because of cloudiness
mol_weight.pop(2)
print(mol_weight)
plt.plot(mol_weight[0:3], t_mtz[0:3], 'ko')
plt.plot(mol_weight[3:6], t_mtz[3:6], 'o')
plt.legend(['concentration of 50 mg/L', 'concentration of 100 mg/L'])
plt.ylabel(r'time to breakthrough [hydraulic residence times]')
plt.xlabel(r'Dye molecular weight [g/mol]')
plt.savefig('/Users/ErinHealy/Desktop/CEE_4530/Final_project/weight_v_tmtz.png', bbox_inches = 'tight')
plt.show()

# Calculate the retardation coefficient $R_{adsorption}$ based on the time to breakthrough for the
R_ads = []
for i in range(np.size(t_mtz)):
    R_ads.append(t_mtz[i]/t_water[i])
print(R_ads)
print(t_mtz)

# Calculate the $q_0$ for each of the columns based on equation (97). Plot this as a function of t
q_0 = []
for i in range(np.size(R_ads)):
    Q = (R_ads[i]-1)*C_0[i]*porosity*Column_V/(M[i]*u.g)
    q_0.append(Q*(1*u.l*u.g/(1000*u.ml*1000*u.mg))) # units removed for graphing purposes. unit is
print(q_0)

```

