

Comparative microfluidic screening of amino acid salt solutions for post-combustion CO₂ capture

Alexander P. Hallenbeck^{a,b}, Adefemi Egbebi^{a,c}, Kevin P. Resnik^{a,c}, David Hopkinson^a, Shelley L. Anna^b, John R. Kitchin^{a,b,*}

^a*National Energy Technology Laboratory, Pittsburgh Pennsylvania, 15236*

^b*Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213*

^c*AECOM, Pittsburgh, PA, 15236*

Contents

1	Summary of Supporting Information	2
2	Gas plug and liquid slug volume calculation	3
3	Precision in concentration calculation	4
4	Gas plug length vs. distance	4
4.1	MEA	5
4.2	TAU	6
4.3	PRO	7
4.4	LYS	8
5	Figures in the manuscript	9
5.1	Figure 3	10
5.2	Figure 4	10

*Corresponding author

Email address: jkitchin@andrew.cmu.edu (John R. Kitchin)

5.3	Figure 5	11
5.4	Figure 6	12
5.5	Figure 7	13
5.6	Figure 8	16
5.7	Figure 9	19
5.8	Figure 10	22
5.9	Figure 11	25

1. Summary of Supporting Information

The supporting information files for this work are meant to provide transparency in how the absorbed CO₂ concentration data presented in this work was derived from images of absorption in the microfluidic reactor. Consequently the raw video files used for each solvent have been provided separately along with Excel sheets containing the raw data. The individual gas plug that was analyzed in each video breaks off from the y-junction between the first and second frame of each video. The excel file for each experiment includes the frame number from which each individual data point was determined. The pixel to micron conversion factor was determined by a separate image of a known 2mm standard image using the same zoom as used for the microfluidic reactor images and the conversion factor is included in each Excel file. The scripts in this document were written in Python.

In addition to the raw video files and data, a summary of the methods used to convert the gas plug length to a volume and then to a concentration of absorbed CO₂ is included below. As the gas plug length data was included for the case of GLY in the main body of this work, the gas plug length

along with the initial gas plug length extrapolation is included below for the remaining solvents.

The source for this document is available here: .

2. Gas plug and liquid slug volume calculation

An example gas plug length measurement in imageJ can be seen below in Figure 1. The line tool in imageJ is used to draw along the path of the centerline of the plug and the measure function is used to return the length, in pixels, of the specified path. The length in pixels is then converted to microns by calibrating with a 2 mm standard image.

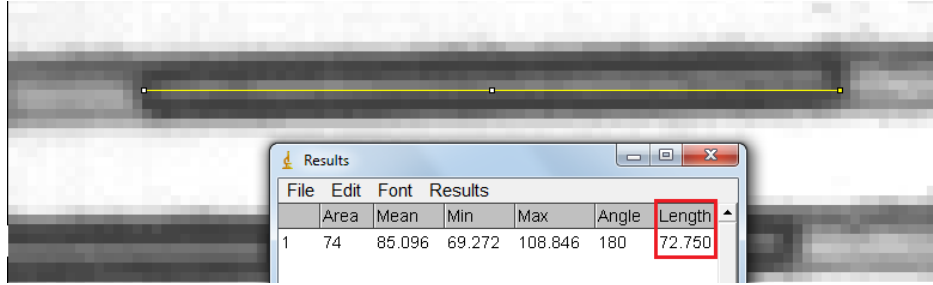


Figure 1: Example of Plug length measurement

As was mentioned in the main body, the hydraulic diameter for the microreactor used in this work was determined in a previous work [1] to be 125 μm . The hydraulic diameter of a gas plug was then assumed to be 121 μm consistent with a liquid film thickness of 1.6% of the channel thickness. The volume of a liquid slug and a gas plug were then calculated using equations 1 and 2 where L_s is the liquid slug length, $r_{h,ls}$ is the hydraulic radius of microchannel, L_g is the gas plug length, $r_{h,gp}$ is the hydraulic radius of the gas plug, and $d_{h,gp}$ is the hydraulic diameter of the gas plug.

$$V_{ls} = L_s \cdot \pi \cdot r_{h,ls}^2 \quad (1)$$


$$V_{gp} = \frac{4}{3} \pi r_{h,gp}^3 + (L_g - d_{h,gp}) \cdot \pi \cdot r_{h,gp}^2 \quad (2)$$

3. Precision in concentration calculation

The gas plug length measurement with imageJ carries with it an estimated uncertainty of +/- 1 pixel or less or equivalently +/- 22-23 μm . Based on Equations 1 and 2, this uncertainty can be propagated to determine the uncertainty of the plug volume and liquid slug volume, which are 0.26 nL and 0.28 nL respectively. The uncertainty in the molar amount of CO_2 can then be calculated using ideal gas law to be approximately 0.018 nmol. Consequently the uncertainty in the concentration measurement in mol/L is +/- ~0.01 M based on Equation 3 and taking a typical slug volume to be 5 nL and a typical molar amount to be 10^{-11} mols.

$$\frac{\delta(x/y)}{(x/y)} = \frac{\delta(x)}{x} + \frac{\delta(y)}{y} \quad (3)$$

4. Gas plug length vs. distance

In this section we plot a representative example of the gas plug lengths vs distance for each solvent, and the exponential fit we obtained for the initial data points. The data used is stored in an embedded Excel sheet, including the exponential fits. GLY is shown in the manuscript, and is not included here. The Excel sheet containing the data used in these scripts can be found here: .

4.1. MEA

```
1 import xlrd
2 import matplotlib.pyplot as plt
3 wb = xlrd.open_workbook('RawData.xlsx')
4 sh1 = wb.sheet_by_name(u'MEA')
5 x1 = sh1.col_values(6,start_rowx=1,end_rowx=44) # distance
6 y1 = sh1.col_values(4,start_rowx=1,end_rowx=44) # plug length
7 x2 = sh1.col_values(15,end_rowx=8) # distanceforfit
8 # pluglengthforfit; Fit being  $y = 1902.853 \cdot \exp((-1.099814 \cdot (10^{-4})) \cdot x)$ 
9 y2 = sh1.col_values(16,end_rowx=8)
10
11 plt.plot(x1, y1, 'ro', x2, y2)
12 plt.xlabel('Distance (microns)')
13 plt.ylabel('Plug Length (microns)')
14 plt.savefig('MEApluglengths.png')
15 plt.show()
```

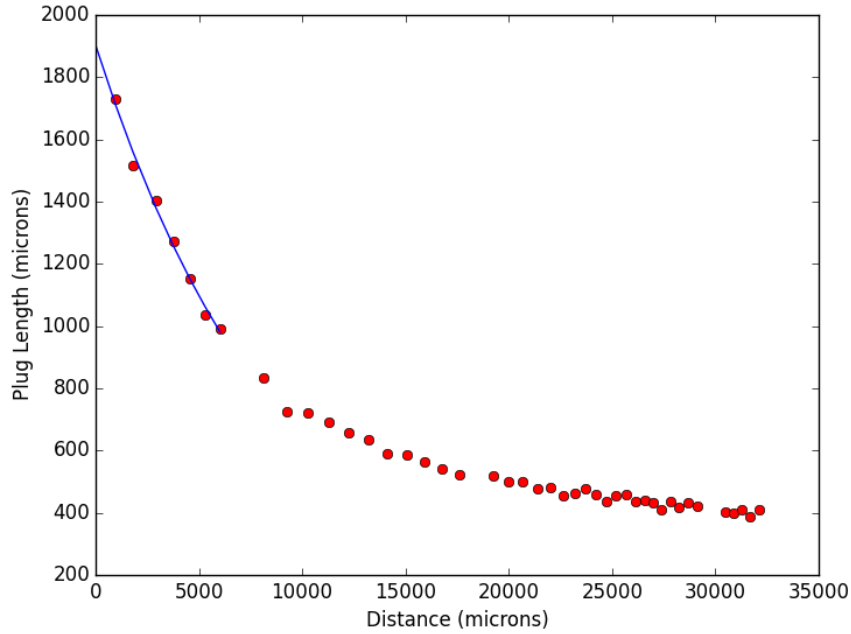


Figure 2: Plug length vs distance used for analysis of 0.50 M MEA with exponential fit for determining initial plug length. Initial Gas plug length determined to be 1903 μm

4.2. TAU

```

1  import xlrd
2  import matplotlib.pyplot as plt
3  wb = xlrd.open_workbook('RawData.xlsx')
4  sh1 = wb.sheet_by_name(u'TAU')
5  x1 = sh1.col_values(6, start_rowx=1, end_rowx=37) # distance
6  y1 = sh1.col_values(4, start_rowx=1, end_rowx=37) # plug length
7  x2 = sh1.col_values(15, end_rowx=11) # distance for fit
8  y2 = sh1.col_values(16, end_rowx=11)
9
10 plt.plot(x1, y1, 'ro', x2, y2)
11 plt.xlabel('Distance (microns)')
12 plt.ylabel('Plug Length (microns)')
13 plt.savefig('TAUpluglengths.png')

```

14 `plt.show()`

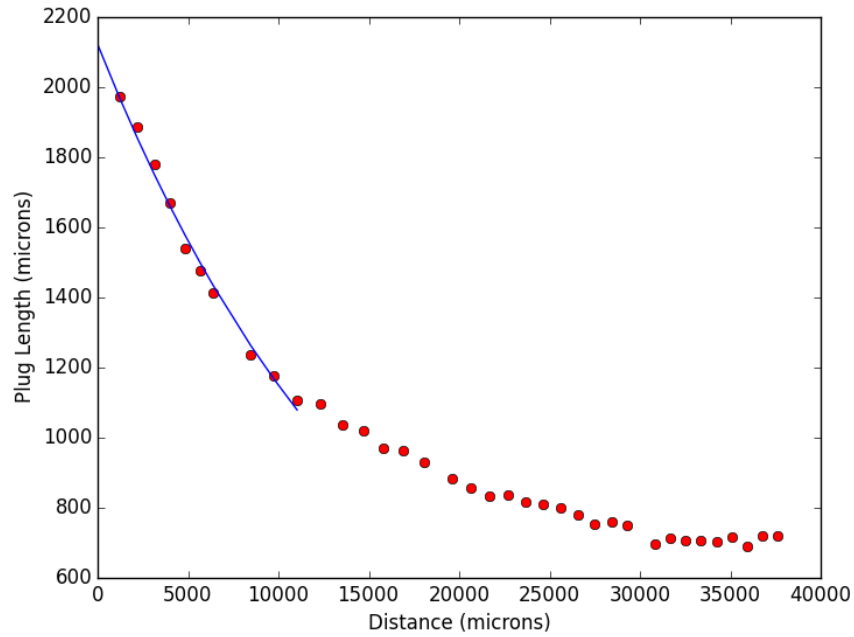


Figure 3: Plug length vs distance used for analysis of 0.50 M K^+TAU^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be 2122 μm

4.3. PRO

```
1 import xlrd
2 import matplotlib.pyplot as plt
3 wb = xlrd.open_workbook('RawData.xlsx')
4 sh1 = wb.sheet_by_name(u'PRO')
5 x1 = sh1.col_values(6, start_rowx=1, end_rowx=34) # distance
6 y1 = sh1.col_values(4, start_rowx=1, end_rowx=34) # plug length
7 x2 = sh1.col_values(15, end_rowx=8) # distance for fit
8 y2 = sh1.col_values(16, end_rowx=8)
9
10 plt.plot(x1, y1, 'ro', x2, y2)
```

```

11 plt.xlabel('Distance (microns)')
12 plt.ylabel('Plug Length (microns)')
13 plt.savefig('PROpluglengths.png')
14 plt.show()

```

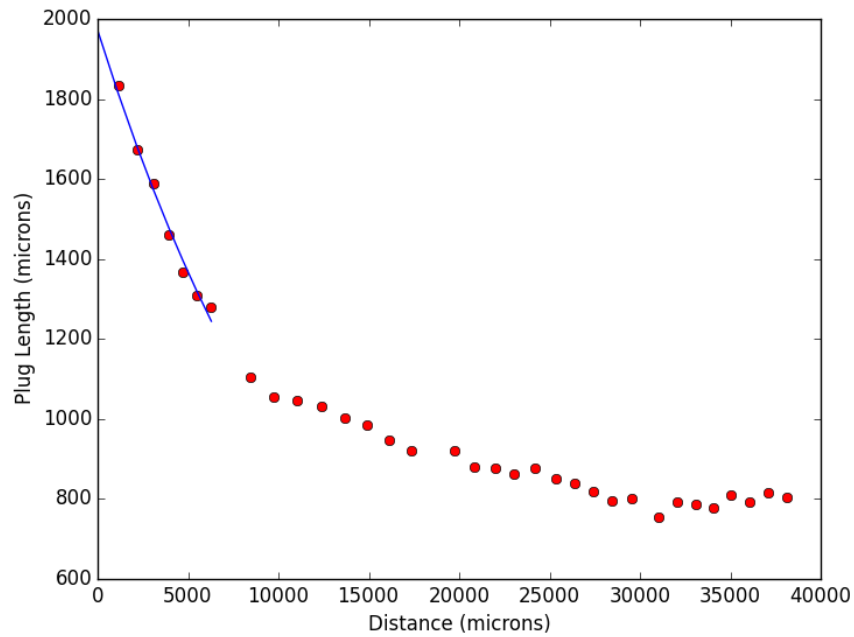


Figure 4: Plug length vs distance used for analysis of 0.50 M K^+PRO^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be $1972 \mu\text{m}$

4.4. LYS

```

1 import xlrd
2 import matplotlib.pyplot as plt
3 wb = xlrd.open_workbook('RawData.xlsx')
4 sh1 = wb.sheet_by_name(u'LYS')
5 x1 = sh1.col_values(6, start_rowx=1, end_rowx=60) # distance
6 y1 = sh1.col_values(4, start_rowx=1, end_rowx=60) # plug length
7 x2 = sh1.col_values(15, end_rowx=8) # distanceforfit

```



```

8  y2 = sh1.col_values(16, end_rowx=8)
9
10 plt.plot(x1, y1, 'ro', x2, y2)
11 plt.xlabel('Distance (microns)')
12 plt.ylabel('Plug Length (microns)')
13 plt.savefig('LYSpluglengths.png')
14 plt.show()

```

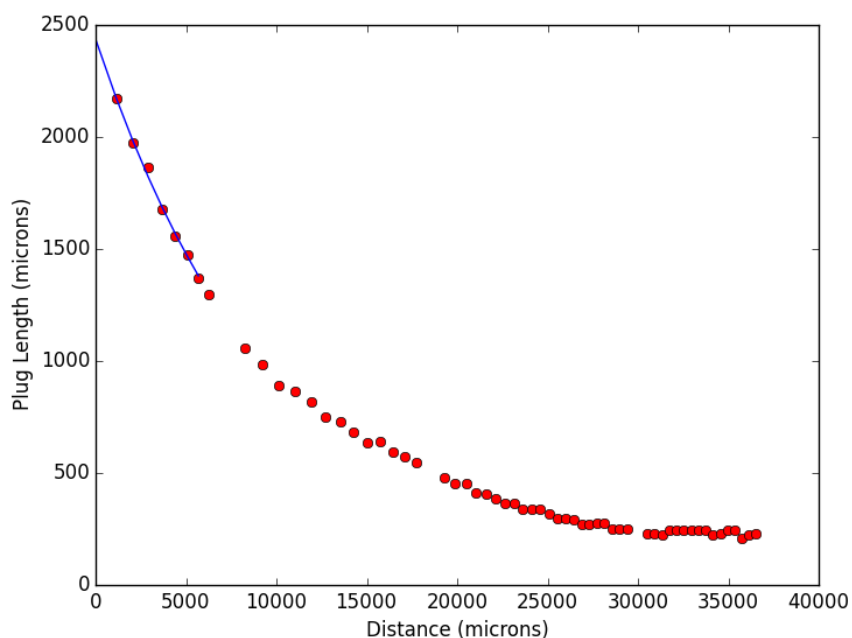


Figure 5: Plug length vs distance used for analysis of 0.50 M K^+LYS^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be 2436 μm

5. Figures in the manuscript

The data used in these figures is also embedded in the attached file Raw-Data.xlsx.

5.1. Figure 3

```
1 import xlrd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from matplotlib.ticker import MultipleLocator, FormatStrFormatter
5
6 majorLocator = MultipleLocator(20)
7 majorFormatter = FormatStrFormatter('%d')
8
9 wb = xlrd.open_workbook('RawData.xlsx')
10 sh1 = wb.sheet_by_name(u'GLY')
11 x1 = np.array(sh1.col_values(6,start_rowx=1,end_rowx=37)) # distance
12 y1 = sh1.col_values(4,start_rowx=1,end_rowx=37) # plug length
13 x2 = np.array(sh1.col_values(15,end_rowx=9)) # distanceforfit
14
15 # pluglengthforfit; Fit being  $y = 2275.629 \cdot \exp((-5.485365 \cdot (10^{-5})) \cdot x)$ 
16 y2 = sh1.col_values(16,end_rowx=9)
17
18 plt.figure(figsize=(3, 4))
19 plt.plot(x1 / 1000, y1, 'ro', x2 / 1000, y2)
20 plt.xlabel('Distance (mm)')
21 plt.ylabel('Plug Length (microns)')
22 plt.tight_layout()
23
24 ax = plt.gca()
25 ax.xaxis.set_major_locator(majorLocator)
26 ax.xaxis.set_major_formatter(majorFormatter)
27
28 for ext in ['.png', '.eps', '.pdf']:
29     plt.savefig('../figures/GLYpluglengths' + ext, dpi=300)
30 plt.show()
```

5.2. Figure 4

```
1 import xlrd
2 wb = xlrd.open_workbook('RawData.xlsx')
3 sh2 = wb.sheet_by_name(u'MEA')
```

```

4  sh3 = wb.sheet_by_name(u'GLY')
5  sh4 = wb.sheet_by_name(u'TAU')
6  sh5 = wb.sheet_by_name(u'PRO')
7  sh6 = wb.sheet_by_name(u'LYS')
8  tm = sh2.col_values(9, start_rowx=1, end_rowx=44)
9  cm = sh2.col_values(11, start_rowx=1, end_rowx=44) # MEA data
10 tg = sh3.col_values(9, start_rowx=1, end_rowx=37)
11 cg = sh3.col_values(11, start_rowx=1, end_rowx=37) # GLY data
12 tt = sh4.col_values(9, start_rowx=1, end_rowx=36)
13 ct = sh4.col_values(11, start_rowx=1, end_rowx=36) # TAU data
14 tp = sh5.col_values(9, start_rowx=1, end_rowx=33)
15 cp = sh5.col_values(11, start_rowx=1, end_rowx=33) # PRO data
16 tl = sh6.col_values(9, start_rowx=1, end_rowx=59)
17 cl = sh6.col_values(11, start_rowx=1, end_rowx=59) # LYS data
18
19 import matplotlib.pyplot as plt
20 plt.figure(figsize=(3, 5))
21 plt.plot(tl, cl, color='#FF8000', marker='D', label='LYS')
22 plt.plot(tm, cm, color='r', marker='o', label='MEA')
23 plt.plot(tg, cg, color='g', marker='v', label='GLY')
24 plt.plot(tt, ct, color='m', marker='^', label='TAU')
25 plt.plot(tp, cp, color='b', marker='s', label='PRO')
26 plt.legend(loc='bottom right')
27 plt.xlabel('Time (s)')
28 plt.ylabel('Concentration of absorbed CO$_{2}$ (M)')
29
30 plt.tight_layout()
31 for ext in ['.png', '.eps', '.pdf']:
32     plt.savefig('../figures/summaryplot' + ext, dpi=300)
33
34 plt.show()

```

5.3. Figure 5

```

1  import xlrd
2  wb = xlrd.open_workbook('RawData.xlsx')
3  sh1 = wb.sheet_by_name(u'CSTRLowLYS conc.')
4  sh2 = wb.sheet_by_name(u'CSTRLowMEA conc.')

```

```

5
6 x = sh1.col_values(1,start_rowx=1) #Time
7 y = sh1.col_values(7,start_rowx=1) #lysine co2 loading
8 x1 = sh2.col_values(1,start_rowx=1)
9 y1 = sh2.col_values(6,start_rowx=1) #MEa co2 loading
10
11 import matplotlib.pyplot as plt
12 plt.figure(figsize=(3, 4))
13 plt.plot(x, y, label='K+LYS-')
14 plt.plot(x1, y1, label='MEA')
15 plt.legend(loc='best')
16 plt.xlabel('Reaction Time (min)')
17 plt.ylabel('CO2 loading (mol CO2/mol lysine/MEA)')
18
19 from matplotlib.ticker import MultipleLocator, FormatStrFormatter
20
21 majorLocator = MultipleLocator(100)
22 majorFormatter = FormatStrFormatter('%d')
23
24 ax = plt.gca()
25 ax.xaxis.set_major_locator(majorLocator)
26 ax.xaxis.set_major_formatter(majorFormatter)
27
28 plt.tight_layout()
29 for ext in ['.png', '.eps', '.pdf']:
30     plt.savefig('../figures/lowcstrcompar' + ext, dpi=300)
31
32 plt.show()

```

5.4. Figure 6

```

1 import xlrd
2 wb = xlrd.open_workbook('RawData.xlsx')
3 sh1 = wb.sheet_by_name(u'CSTR-HighLYS conc.')
4 sh2 = wb.sheet_by_name(u'CSTR-HighMEA conc.')
5
6 x = sh1.col_values(10,start_rowx=1) # Time
7 y = sh1.col_values(7,start_rowx=1) # lysine co2 loading

```

```

8 x1 = sh2.col_values(1,start_rowx=1)
9 y1 = sh2.col_values(7,start_rowx=1) # MEA co2 loading
10
11 import matplotlib.pyplot as plt
12 plt.figure(figsize=(3, 4))
13 plt.plot(x, y, label='K$^{+}$LYS$^{-}$')
14 plt.plot(x1, y1, label='MEA')
15 plt.legend(loc='best')
16 plt.xlabel('t*(Q/V) (unitless)')
17 plt.ylabel('CO$_{2}$ loading (mmol CO$_{2}$ /g solution)')
18
19 plt.tight_layout()
20 for ext in ['.png', '.eps', '.pdf']:
21     plt.savefig('../figures/highcstrcompar' + ext, dpi=300)
22
23 plt.show()

```

5.5. Figure 7

```

1 import xlrd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mping
5 from matplotlib.offsetbox import OffsetImage, AnnotationBbox
6
7 wb = xlrd.open_workbook('RawData.xlsx')
8 sh1 = wb.sheet_by_name(u'MEARaman')
9 xf = sh1.col_values(0)
10 yf = sh1.col_values(2) # Raman intensity postMF fingerprint region
11 xpf = sh1.col_values(3)
12 ypf = sh1.col_values(4) # Raman intensity preMF fingerprint region
13 xh = sh1.col_values(6, end_rowx=2583)
14 yh = sh1.col_values(7, end_rowx=2583) # Raman intensity postMF high region
15 xph = sh1.col_values(9, end_rowx=2583)
16 yph = sh1.col_values(10, end_rowx=2583) # Raman intensity preMF high region
17
18 # data has been read in
19 fig = plt.figure()

```

```

20
21 # figure created
22 ax = fig.add_subplot(1, 2, 1)
23
24 # left subplot added
25 ax.plot(xpf, ypf, label='Pre-MF expt') # plotting pre data
26
27 xpf = np.array(xpf)
28 ypf = np.array(ypf)
29 ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
30 shadeu = (xpf > 1400) & (xpf < 1600)
31 ax.set_ylim(20, 100)
32 ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
33                y2=ymax*np.ones(len(xpf[shadeu])),
34                color='#C8C8C8')
35 shadei = (xpf > 800) & (xpf < 1000)
36
37 ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
38                y2=ymax * np.ones(len(xpf[shadei])), color='#B8B8B8')
39
40 ax.plot(xf, yf, label='Post-MF expt') # plotting post data
41 ax.set_xlim([800, 1800])
42 ax.legend(loc='upper right')
43 ax.set_xlabel('Wavenumber (cm-1)')
44 ax.set_ylabel('Raman Intensity (a.u.)')
45 ax.set_yticklabels([])
46
47 ax.text(1450, 30, r'CO2-')
48 ax.text(1420, 27, r'Stretch')
49 ax.text(1130, 76, r'1162')
50 ax.text(1145, 79, r'$\bigtriangleup$', color='r')
51 ax.text(1030, 92, r'1067')
52 ax.text(1045, 95, r'$\bigcirc$', color='r')
53 ax.text(910, 90, r'1017')
54 ax.text(950, 92.5, r'$\bigstar$', color='r')
55 ax.text(850, 35, r'CH2')
56 ax.text(835, 32, r'Twist')
57 ax.text(840, 29, r'CCO')

```

```

58 ax.text(818, 26, r'Stretch')
59 ax.text(1020, 33, r'$\bigtriangleup$', color='r')
60 ax.text(1080, 33, r'Carbamate', color='r')
61 ax.text(1020, 30, r'$\bigstar$', color='r')
62 ax.text(1080, 29, r'HCO$_{3}$^{-}$', color='r')
63 ax.text(1020, 26, r'$\bigcirc$', color='r')
64 ax.text(1080, 25, r'CO$_{3}$^{-2}$', color='r')
65 # plot with shading
66 ax = fig.add_subplot(1, 2, 2)
67
68 # right subplot added
69 ax.plot(xph, yph, label='Pre-MF expt')
70 xph = np.array(xph)
71 yph = np.array(yph)
72 ymax = ax.get_ylim()[1]
73 shadex = (xph > 3200) & (xph < 3400)
74 ax.fill_between(xph[shadex], y1=np.zeros(len(xph[shadex])),
75                y2=ymax*np.ones(len(xph[shadex])),
76                color='#D0D0D0')
77 shadeq = (xph>2800) & (xph< 3000)
78 ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
79                y2=ymax*np.ones(len(xph[shadeq])), color='#A8A8A8')
80 ax.plot(xh, yh, label='Post-MF expt')
81 ax.set_xlim([2800, 3800])
82
83 ax.set_xlabel('Wavenumber (cm$^{-1}$)')
84 ax.set_yticklabels([])
85
86 # plotting with shading
87 image = mpimg.imread('800px-Ethanolamine-2D-skeletal-B.png')
88
89 # read in image and make into array of colors
90 imagebox=OffsetImage(image, zoom=0.15)
91 ax.text(3250, 300, r'NH$_{2}$')
92 ax.text(3250, 275, r'N-H')
93 ax.text(3210, 250, r'Stretch')
94 ax.text(3170, 570, r'3317')
95 ax.text(2850, 300, r'CH$_{2}$')

```

```

96 ax.text(2850, 275, r'C-H')
97 ax.text(2810, 250, r'Stretch')
98
99 # rescaling size of image
100 structure = AnnotationBbox(imagebox, xy=(3300, 50), frameon=False)
101
102 # places image on plot at xy location
103 ax.add_artist(structure)
104 plt.tight_layout()
105
106 for ext in ['.png', '.eps', '.pdf']:
107     plt.savefig('../figures/MEAMFRamanwithstructure' + ext, dpi=300)
108
109 plt.show()

```

5.6. Figure 8

Raman spectra for GLY.

```

1 import xlrd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mping
5 from matplotlib.offsetbox import OffsetImage, AnnotationBbox
6
7 #reading in data
8 wb = xlrd.open_workbook('RawData.xlsx')
9 sh1=wb.sheet_by_name(u'GlyRaman')
10 xf=sh1.col_values(3) #wavenumber
11 yf=sh1.col_values(5) #Raman intensity postMF fingerprint region
12 xpf=sh1.col_values(0)
13 ypf=sh1.col_values(1) # Raman intensity preMF fingerprint region
14 xh=sh1.col_values(9, end_rowx=2583)
15 yh=sh1.col_values(10, end_rowx=2583) #Raman intensity postMF high region
16 xph=sh1.col_values(6, end_rowx=2583)
17 yph=sh1.col_values(7, end_rowx=2583) #Raman intensity preMF high region
18

```



```

19 # creating Figure
20 fig=plt.figure()
21
22 # adding left subplot
23 ax = fig.add_subplot(1, 2, 1)
24 ax.plot(xpf, ypf, label='Pre-MF expt') # plotting pre data
25
26 # making shading
27 xpf = np.array(xpf)
28 ypf = np.array(ypf)
29 shadeu = (xpf > 1400) & (xpf < 1600)
30 ax.set_ylim(5, 100)
31 ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
32 ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
33                y2=ymax*np.ones(len(xpf[shadeu])),
34                color='#C8C8C8')
35 shadei= (xpf>800) & (xpf<1000)
36
37 ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
38                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
39 ax.plot(xf, yf, label='Post-MF expt')#plotting post data
40 ax.set_xlim([800, 1800])
41 ax.legend(loc='upper right')
42
43 ax.text(1450, 14, r'CO$_{2}$^{+}$')
44 ax.text(1410, 10, r'Stretch')
45 ax.text(1430, 57, r'1445')
46 ax.text(1135, 51, r'1175')
47 ax.text(1140, 54, r'$\bigtriangleup$', color='r')
48 ax.text(990, 71, r'1045')
49 ax.text(1010, 74, r'$\bigstar$', color='r')
50 ax.text(1000, 77, r'$\bigcirc$', color='r')
51 ax.text(900, 80, r'914')
52 ax.text(850, 22, r'CH$_{2}$')
53 ax.text(835, 19, r'Twist')
54 ax.text(850, 16, r'CCO')
55 ax.text(820, 13, r'Stretch')
56 ax.text(1500, 83, r'Carbamate', color='r')

```

```

57 ax.text(1430, 83, r'$\bigtriangleup$', color='r')
58 ax.text(1440, 79, r'$\bigstar$', color='r')
59 ax.text(1500, 79, r'HCO$_{3}$$^{-}$', color='r')
60 ax.text(1430, 75, r'$\bigcirc$', color='r')
61 ax.text(1500, 75, r'CO$_{3}$$^{-2}$', color='r')
62 ax.set_xlabel('Wavenumber (cm$^{-1}$)')
63 ax.set_ylabel('Raman Intensity (a.u.)')
64 ax.set_yticklabels([])
65
66 # adding subplot two
67 ax = fig.add_subplot(1, 2, 2)
68 ax.plot(xph, yph, label='Pre-MF expt')
69
70 # making shading
71 xph = np.array(xph)
72 yph = np.array(yph)
73 ax.set_ylim(0, 700)
74 ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
75 shadex = (xph > 3200) & (xph < 3400)
76 ax.fill_between(xph[shadex], y1=np.zeros(len(xph[shadex])),
77                y2=ymax*np.ones(len(xph[shadex])),
78                color='#D0D0D0')
79 shadeq = (xph>2800) & (xph< 3000)
80 ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
81                y2=ymax*np.ones(len(xph[shadeq])), color='#A8A8A8')
82 ax.plot(xh, yh, label='Post-MF expt')
83 ax.set_xlim([2800, 3800])
84
85 ax.set_xlabel('Wavenumber (cm$^{-1}$)')
86 ax.set_yticklabels([])
87
88 # text+image
89 ax.text(3250, 400, r'NH$_{2}$')
90 ax.text(3250, 370, r'N-H')
91 ax.text(3210, 345, r'Stretch')
92 ax.text(3245, 556, r'3320')
93 ax.text(2850, 400, r'CH$_{2}$')
94 ax.text(2850, 370, r'C-H')

```

```

95 ax.text(2810, 345, r'Stretch')
96 image = mpimg.imread('glycine.png')
97
98 # read in image and make into array of colors
99 imagebox=OffsetImage(image, zoom=0.8)
100
101 # rescaling size of image
102 structure = AnnotationBbox(imagebox, xy=(3300, 100), frameon=False)
103
104 # places image on plot at xy location
105 ax.add_artist(structure)
106 plt.tight_layout()
107
108 for ext in ['.png', '.eps', '.pdf']:
109     plt.savefig('../figures/GLYMFramanwithstructure.png', dpi=300)
110 plt.show()

```

5.7. Figure 9

```

1 import xlrd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mpimg
5 from matplotlib.offsetbox import OffsetImage, AnnotationBbox
6
7 #reading in data
8 wb = xlrd.open_workbook('RawData.xlsx')
9 sh1 = wb.sheet_by_name(u'TauRaman')
10 xpf = sh1.col_values(0) # wavenumber
11 ypf = sh1.col_values(1) # Raman intensity preMF fingerprint region
12 xf = sh1.col_values(3)
13 yf = sh1.col_values(5) # Raman intensity postMF fingerprint region
14 xph = sh1.col_values(6, end_rowx=2583)
15 yph = sh1.col_values(7, end_rowx=2583) # Raman intensity preMF high region
16 xh = sh1.col_values(9, end_rowx=2583)
17 yh = sh1.col_values(10, end_rowx=2583) # Raman intensity postMF high region
18
19 # Creating figure

```

```

20 fig = plt.figure()
21
22 # adding left subplot
23 ax = fig.add_subplot(1, 2, 1)
24 ax.plot(xpf, ypf, label='Pre-MF expt') # plotting pre data
25
26 # making shading
27 xpf = np.array(xpf)
28 ypf = np.array(ypf)
29 shadeu = (xpf > 1400) & (xpf < 1600)
30 ax.set_ylim(0, 400)
31 ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
32 ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
33                y2=ymax*np.ones(len(xpf[shadeu])),
34                color='#C8C8C8')
35 shadei= (xpf>700) & (xpf<1000)
36
37 ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
38                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
39 shadee= (xpf>1030) & (xpf<1070)
40 ax.fill_between(xpf[shadee], y1=np.zeros(len(xpf[shadee])), y2=ymax*np.ones(len(xpf[shadee])), color='#COCOCO')
41
42 ax.plot(xf, yf, label='Post-MF expt') #plotting post data
43 ax.set_xlim([700, 1700])
44 ax.legend(loc='upper right')
45 ax.text(1450, 100, r'C0$_{2}$^{--}$')
46 ax.text(1410, 85, r'Stretch')
47 ax.text(1280, 60, r'1340')
48 ax.text(1310, 72, r'$\bigstar$', color='r')
49 ax.text(1020, 250, r'S0$_{3}$^{--}$')
50 ax.text(1020, 230, r'Stretch')
51 ax.text(900, 80, r'955')
52 ax.text(800, 300, r'CH$_{2}$')
53 ax.text(800, 285, r'Twist')
54 ax.text(800, 270, r'CCO')
55 ax.text(800, 255, r'Stretch')
56 ax.text(1530, 300, r'HCO$_{3}$^{--}$', color='r')
57 ax.text(1480, 300, r'$\bigstar$', color='r')

```

```

58 ax.set_xlabel('Wavenumber (cm-1)')
59 ax.set_ylabel('Raman Intensity (a.u.)')
60 ax.set_yticklabels([])
61
62 # adding second subplot
63 ax = fig.add_subplot(1, 2, 2)
64 ax.plot(xph, yph, label='Pre-MF expt')
65
66 # making shading
67 xph = np.array(xph)
68 yph = np.array(yph)
69 ax.set_ylim(0, 700)
70 ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
71 shadex = (xph > 3200) & (xph < 3400)
72 ax.fill_between(xph[shadex], y1=np.zeros(len(xph[shadex])),
73                y2=ymax*np.ones(len(xph[shadex])),
74                color='#D0D0D0')
75 shadeq = (xph>2800) & (xph< 3000)
76 ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
77                y2=ymax*np.ones(len(xph[shadeq])),
78                color='#A8A8A8')
79 ax.plot(xh, yh, label='Post-MF expt')
80 ax.set_xlim([2800, 3800])
81
82 ax.set_xlabel('Wavenumber (cm-1)')
83 ax.set_yticklabels([])
84 #text+image
85 ax.text(3250, 300, r'NH2')
86 ax.text(3250, 270, r'N-H')
87 ax.text(3210, 250, r'Stretch')
88 ax.text(3270, 560, r'3317')
89 ax.text(2850, 300, r'CH2')
90 ax.text(2850, 270, r'C-H')
91 ax.text(2810, 250, r'Stretch')
92
93 image = mpimg.imread('Taurine.png')
94
95 # read in image and make into array of colors

```

```

96 imagebox=OffsetImage(image, zoom=0.4)
97
98 # rescaling size of image
99 structure = AnnotationBbox(imagebox, xy=(3300, 90), frameon=False)
100
101 # places image on plot at xy location
102 ax.add_artist(structure)
103 plt.tight_layout()
104
105 for ext in ['.png', '.eps', '.pdf']:
106     plt.savefig('../figures/TauMFRamanwithstructure' + ext, dpi=300)
107
108 plt.show()

```

5.8. Figure 10

```

1 import xlrd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mpimg
5 from matplotlib.offsetbox import OffsetImage, AnnotationBbox
6
7 # reading in data
8 wb = xlrd.open_workbook('RawData.xlsx')
9 sh1 = wb.sheet_by_name(u'ProRaman')
10 xpf = sh1.col_values(0) # wavenumber
11 ypf = sh1.col_values(1) # Raman intensity preMF fingerprint region
12 xf = sh1.col_values(3)
13 yf = sh1.col_values(5) # Raman intensity postMF fingerprint region
14 xph = sh1.col_values(9, end_rowx=2583)
15 yph = sh1.col_values(10, end_rowx=2583) # Raman intensity preMF high region
16 xh = sh1.col_values(6, end_rowx=2583)
17 yh = sh1.col_values(7, end_rowx=2583) # Raman intensity postMF high region
18
19 # creating Figure
20 fig = plt.figure()
21
22 # adding left subplot

```

```

23 ax = fig.add_subplot(1, 2, 1)
24 ax.plot(xpf, ypf, label='Pre-MF expt') # plotting pre data
25
26 # making shading
27 xpf = np.array(xpf)
28 ypf = np.array(ypf)
29 shadeu = (xpf > 1400) & (xpf < 1600)
30 ax.set_ylim(0, 120)
31 ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
32 ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
33                y2=ymax*np.ones(len(xpf[shadeu])),
34                color='#C8C8C8')
35 shadei= (xpf>800) & (xpf<1000)
36
37 ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
38                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
39 shadeo= (xpf>1130) & (xpf<1190)
40 ax.fill_between(xpf[shadeo], y1=np.zeros(len(xpf[shadeo])),
41                y2=ymax*np.ones(len(xpf[shadeo])), color='#B0B0B0')
42
43 ax.plot(xf, yf, label='Post-MF expt') # plotting post data
44 ax.set_xlim([800, 1800])
45 ax.legend(loc='upper right')
46 ax.text(1450, 16, r'CO$_{2}$^{--}$')
47 ax.text(1410, 14, r'Stretch')
48 ax.text(1410, 8, r'R$_{1}$R$_{2}$NH')
49 ax.text(1450, 4, r'N-H')
50 ax.text(1410, 0.5, r'Stretch')
51 ax.text(1275, 75, r'1350')
52 ax.text(1310, 79, r'$\bigstar$', color='r')
53 ax.text(1070, 14, r'R$_{1}$R$_{2}$NH')
54 ax.text(1093, 10, r'C-N-C')
55 ax.text(1080, 6, r'Stretch')
56 ax.text(850, 20, r'CH$_{2}$')
57 ax.text(840, 16, r'Twist')
58 ax.text(845, 11, r'CCO')
59 ax.text(820, 7, r'Stretch')
60 ax.text(1630, 95, r'HCO$_{3}$^{--}$', color='r')

```

```

61 ax.text(1592, 96, r'$\bigstar$', color='r')
62 ax.set_xlabel('Wavenumber (cm$^{-1}$)')
63 ax.set_ylabel('Raman Intensity (a.u.)')
64 ax.set_yticklabels([])
65
66 # adding subplot two
67 ax = fig.add_subplot(1, 2, 2)
68 ax.plot(xph, yph, label='Pre-MF expt')
69
70 # making shading
71 xph = np.array(xph)
72 yph = np.array(yph)
73 ax.set_ylim(0, 700)
74 ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
75
76 shadeq = (xph > 2800) & (xph < 3000)
77 ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
78                y2=ymax*np.ones(len(xph[shadeq])),
79                color='#A8A8A8')
80 ax.plot(xh, yh, label='Post-MF expt')
81 ax.set_xlim([2800, 3800])
82 ax.text(2850, 400, r'CH$_{2}$')
83 ax.text(2850, 370, r'C-H')
84 ax.text(2810, 350, r'Stretch')
85 ax.set_xlabel('Wavenumber (cm$^{-1}$)')
86 ax.set_yticklabels([])
87
88 # text + image
89 image = mpimg.imread('Proline.png')
90
91 # read in image and make into array of colors
92 imagebox=OffsetImage(image, zoom=0.8)
93
94 # rescaling size of image
95 structure = AnnotationBbox(imagebox, xy=(3300, 100), frameon=False)
96
97 # places image on plot at xy location
98 ax.add_artist(structure)

```



```

99 plt.tight_layout()
100
101 for ext in ['.png', '.eps', '.pdf']:
102     plt.savefig('../figures/PROMFRamanwithstructure' + ext, dpi=300)
103 plt.show()

```

5.9. Figure 11

```

1 import xlrd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.image as mpimg
5 from matplotlib.offsetbox import OffsetImage, AnnotationBbox
6
7 # reading in data
8 wb = xlrd.open_workbook('RawData.xlsx')
9 sh1=wb.sheet_by_name(u'LysRaman')
10 xpf=sh1.col_values(0) # wavenumber
11 ypf=sh1.col_values(1) # Raman intensity preMF fingerprint region
12 xf=sh1.col_values(3)
13 yf=sh1.col_values(5) # Raman intensity postMF fingerprint region
14 xph=sh1.col_values(9, end_rowx=2583)
15 yph=sh1.col_values(10, end_rowx=2583) # Raman intensity preMF high region
16 xh=sh1.col_values(6, end_rowx=2583)
17 yh=sh1.col_values(7, end_rowx=2583) # Raman intensity postMF high region
18
19 # creating Figure
20 fig=plt.figure()
21
22 # adding left subplot
23 ax = fig.add_subplot(1, 2, 1)
24 ax.plot(xpf, ypf, label='Pre-MF expt') # plotting pre data
25
26 # making shading
27 xpf = np.array(xpf)
28 ypf = np.array(ypf)
29 shadeu = (xpf > 1400) & (xpf < 1600)
30 ax.set_ylim(10, 120)

```

```

31  ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
32  ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
33                 y2=ymax*np.ones(len(xpf[shadeu])),
34                 color='#C8C8C8')
35  shadei= (xpf>800) & (xpf<1000)
36
37  ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
38                 y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
39
40  ax.plot(xf, yf, label='Post-MF expt') # plotting post data
41  ax.set_xlim([800, 1800])
42  ax.legend(loc='upper right')
43  ax.text(1450, 25, r'CO$_{2}$^{--}$')
44  ax.text(1410, 21, r'Stretch')
45  ax.text(1100, 80, r'1130')
46  ax.text(1100, 83, r'$\bigtriangleup$', color='r')
47  ax.text(1030, 106, r'1068')
48  ax.text(1030, 110, r'$\bigcirc$', color='r')
49  ax.text(930, 96.5, r'1017')
50  ax.text(992, 100, r'$\bigstar$', color='r')
51  ax.text(850, 40, r'CH$_{2}$')
52  ax.text(830, 36, r'Twist')
53  ax.text(850, 28, r'CCO')
54  ax.text(820, 24, r'Stretch')
55  ax.text(1510, 100, r'Carbamate', color='r')
56  ax.text(1450, 100, r'$\bigtriangleup$', color='r')
57  ax.text(1460, 95, r'$\bigstar$', color='r')
58  ax.text(1510, 95, r'HCO$_{3}$^{--}$', color='r')
59  ax.text(1450, 90, r'$\bigcirc$', color='r')
60  ax.text(1510, 90, r'CO$_{3}$^{-2}$', color='r')
61  ax.set_xlabel('Wavenumber (cm$^{-1}$)')
62  ax.set_ylabel('Raman Intensity (a.u.)')
63  ax.set_yticklabels([])
64
65  # adding subplot two
66  ax = fig.add_subplot(1, 2, 2)
67  ax.plot(xph, yph, label='Pre-MF expt')
68

```

```

69  # making shading
70  xph = np.array(xph)
71  yph = np.array(yph)
72  ax.set_ylim(0, 800)
73  ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
74
75  shadeq = (xph>2800) & (xph< 3000)
76  ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
77                y2=ymax*np.ones(len(xph[shadeq])), color='#A8A8A8')
78  shadey= (xph>3200) & (xph<3400)
79  ax.fill_between(xph[shadey], y1=np.zeros(len(xph[shadey])),
80                y2=ymax*np.ones(len(xph[shadey])), color= '#A0A0A0')
81
82  ax.plot(xh, yh, label='Post-MF expt')
83  ax.set_xlim([2800, 3800])
84  ax.set_xlabel('Wavenumber (cm-1)')
85  ax.set_yticklabels([])
86
87  # text+image
88  ax.text(3250, 400, r'NH2')
89  ax.text(3250, 370, r'N-H')
90  ax.text(3210, 350, r'Stretch')
91  ax.text(3250, 730, r'3310')
92  ax.text(2850, 400, r'CH2')
93  ax.text(2850, 370, r'C-H')
94  ax.text(2810, 350, r'Stretch')
95  image = mpimg.imread('Lysine.png')
96
97  # read in image and make into array of colors
98  imagebox=OffsetImage(image, zoom=0.65)
99
100 # rescaling size of image
101 structure = AnnotationBbox(imagebox, xy=(3320, 75), frameon=False)
102
103 # places image on plot at xy location
104 ax.add_artist(structure)
105 plt.tight_layout()
106

```

```
107 for ext in ['.png', '.eps', '.pdf']:
108     plt.savefig('../figures/LYSMFramanwithstructure' + ext, dpi=300)
109
110 plt.show()
```

References

- [1] Wei Li, Kun Liu, Ryan Simms, Jesse Greener, Dinesh Jagadeesan, Sascha Pinto, Axel Gunther, and Eugenia Kumacheva. Microfluidic study of fast gas-liquid reactions. *Journal of the American Chemical Society*, 134(6):3127–3132, 2012. PMID: 22176612.