Comparative microfluidic screening of amino acid salt solutions for post-combustion CO_2 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

*Corresponding author
Email address: `jkitchin@andrew.cmu.edu` (John R. Kitchin)

1. Summary of Supporting Information

The supporting information files for this work are meant to provide transparency in how the absorbed CO_2 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 CO_2 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.

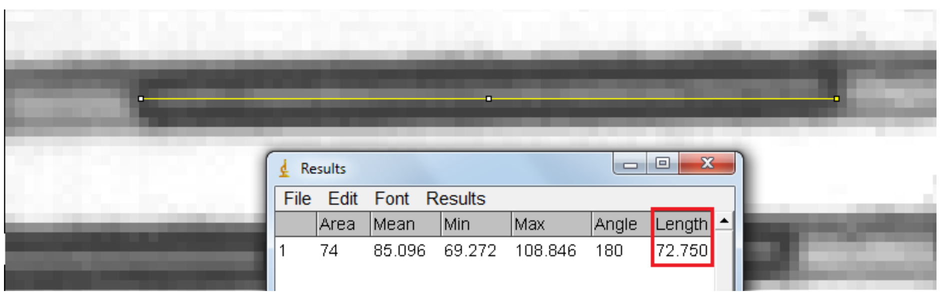

Figure 1: Example of Plug length measurement

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.

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

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

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.

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

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

```
import xlrd
import matplotlib.pyplot as plt
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'MEA')
x1 = sh1.col_values(6,start_rowx=1,end_rowx=44)  # distance
y1 = sh1.col_values(4,start_rowx=1,end_rowx=44)  # plug length
x2 = sh1.col_values(15,end_rowx=8)  # distanceforfit
# pluglengthforfit; Fit being y= 1902.853*exp((-1.099814*(10^-4))*x)
y2 = sh1.col_values(16,end_rowx=8)

plt.plot(x1, y1, 'ro', x2, y2)
plt.xlabel('Distance (microns)')
plt.ylabel('Plug Length (microns)')
plt.savefig('MEApluglengths.png')
plt.show()
```

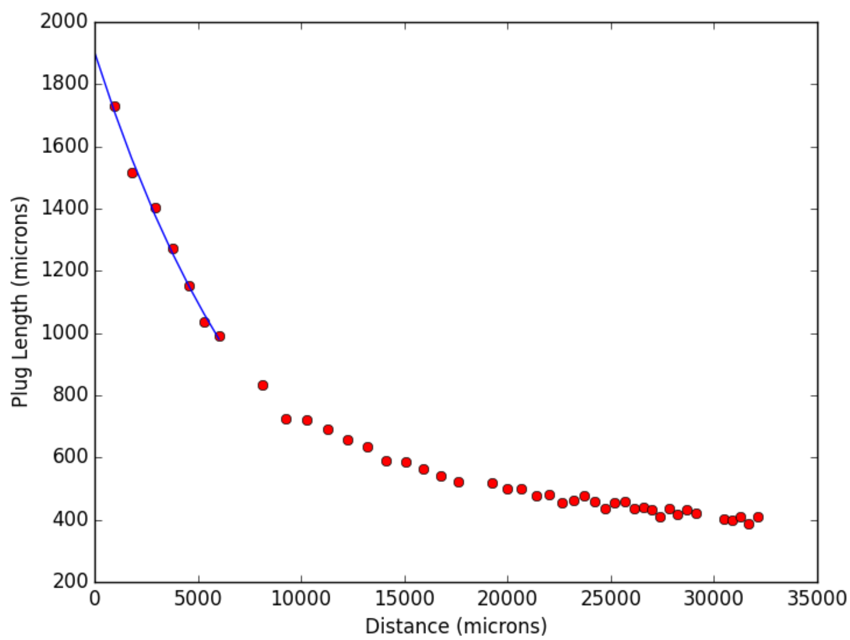

Figure 2: Plug length vs distance used for analysis of 0.50 M MEA with exponential fit for determining initial plug length. Initial Gas plug length determined to be 1903 μm

4.2. TAU

```python
import xlrd
import matplotlib.pyplot as plt
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'TAU')
x1 = sh1.col_values(6, start_rowx=1, end_rowx=37)  # distance
y1 = sh1.col_values(4, start_rowx=1, end_rowx=37)  # plug length
x2 = sh1.col_values(15, end_rowx=11)  # distanceforfit
y2 = sh1.col_values(16, end_rowx=11)

plt.plot(x1, y1, 'ro', x2, y2)
plt.xlabel('Distance (microns)')
plt.ylabel('Plug Length (microns)')
plt.savefig('TAUpluglengths.png')
```

```
plt.show()
```

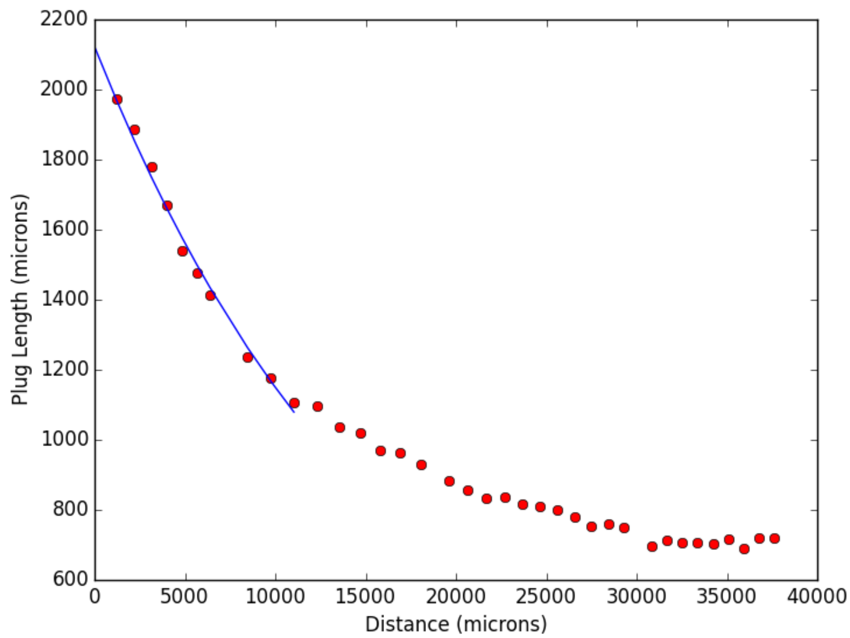

Figure 3: Plug length vs distance used for analysis of 0.50 M K^+TAU^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be 2122 μm

4.3. PRO

```
import xlrd
import matplotlib.pyplot as plt
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'PRO')
x1 = sh1.col_values(6, start_rowx=1, end_rowx=34)  # distance
y1 = sh1.col_values(4, start_rowx=1, end_rowx=34)  # plug length
x2 = sh1.col_values(15, end_rowx=8)  # distanceforfit
y2 = sh1.col_values(16, end_rowx=8)

plt.plot(x1, y1, 'ro', x2, y2)
```

```
plt.xlabel('Distance (microns)')
plt.ylabel('Plug Length (microns)')
plt.savefig('PROpluglengths.png')
plt.show()
```

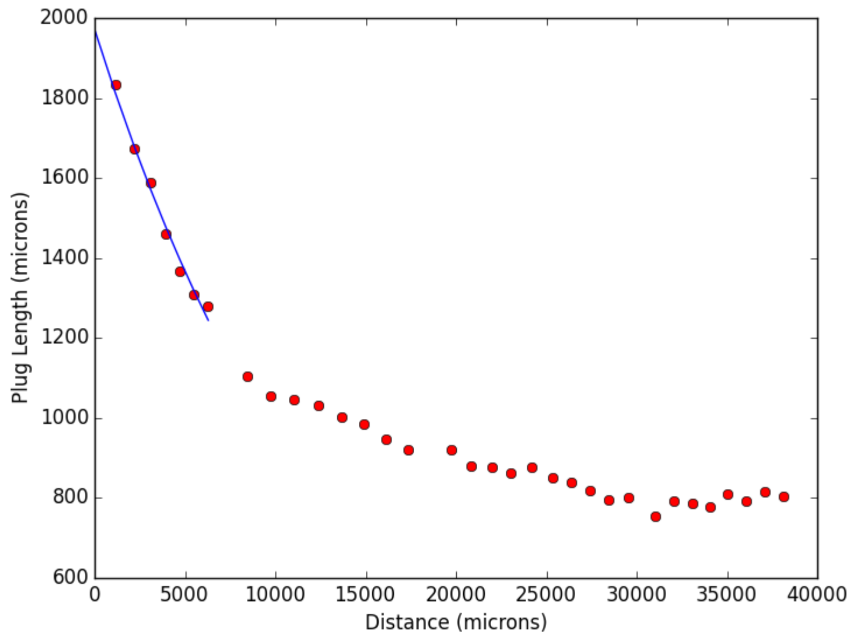

Figure 4: Plug length vs distance used for analysis of 0.50 M K^+PRO^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be 1972 μm

4.4. LYS

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

```
y2 = sh1.col_values(16, end_rowx=8)

plt.plot(x1, y1, 'ro', x2, y2)
plt.xlabel('Distance (microns)')
plt.ylabel('Plug Length (microns)')
plt.savefig('LYSpluglengths.png')
plt.show()
```

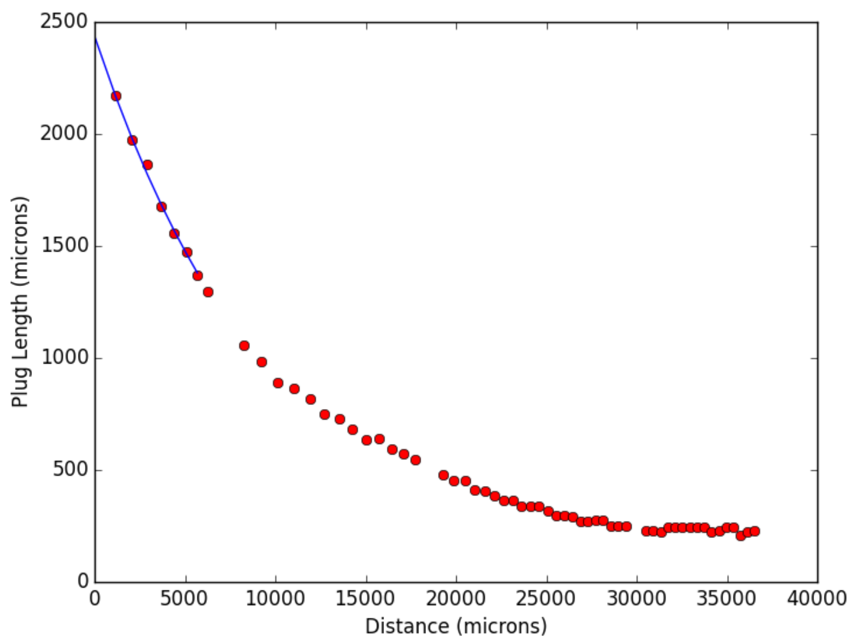

Figure 5: Plug length vs distance used for analysis of 0.50 M K^+LYS^- with exponential fit for determining initial plug length. Initial Gas plug length determined to be 2436 μm

5. Figures in the manuscript

The data used in these figures is also embedded in the attached file RawData.xlsx.

5.1. *Figure 3*

```
import xlrd
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.ticker import MultipleLocator, FormatStrFormatter

majorLocator   = MultipleLocator(20)
majorFormatter = FormatStrFormatter('%d')

wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'GLY')
x1 = np.array(sh1.col_values(6,start_rowx=1,end_rowx=37))  # distance
y1 = sh1.col_values(4,start_rowx=1,end_rowx=37)  # plug length
x2 = np.array(sh1.col_values(15,end_rowx=9))  # distanceforfit

# pluglengthforfit; Fit being y= 2275.629*exp((-5.485365*(10^-5))*x)
y2 = sh1.col_values(16,end_rowx=9)

plt.figure(figsize=(3, 4))
plt.plot(x1 / 1000, y1, 'ro', x2 / 1000, y2)
plt.xlabel('Distance (mm)')
plt.ylabel('Plug Length (microns)')
plt.tight_layout()

ax = plt.gca()
ax.xaxis.set_major_locator(majorLocator)
ax.xaxis.set_major_formatter(majorFormatter)

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

5.2. *Figure 4*

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

```
sh3 = wb.sheet_by_name(u'GLY')
sh4 = wb.sheet_by_name(u'TAU')
sh5 = wb.sheet_by_name(u'PRO')
sh6 = wb.sheet_by_name(u'LYS')
tm = sh2.col_values(9, start_rowx=1, end_rowx=44)
cm = sh2.col_values(11, start_rowx=1, end_rowx=44)  # MEA data
tg = sh3.col_values(9, start_rowx=1, end_rowx=37)
cg = sh3.col_values(11, start_rowx=1, end_rowx=37)  # GLY data
tt = sh4.col_values(9, start_rowx=1, end_rowx=36)
ct = sh4.col_values(11, start_rowx=1, end_rowx=36)  # TAU data
tp = sh5.col_values(9, start_rowx=1, end_rowx=33)
cp = sh5.col_values(11, start_rowx=1, end_rowx=33)  # PRO data
tl = sh6.col_values(9, start_rowx=1, end_rowx=59)
cl = sh6.col_values(11, start_rowx=1, end_rowx=59)  # LYS data

import matplotlib.pyplot as plt
plt.figure(figsize=(3, 5))
plt.plot(tl, cl, color='#FF8000', marker='D', label='LYS')
plt.plot(tm, cm, color='r', marker='o', label='MEA')
plt.plot(tg, cg, color='g', marker='v', label='GLY')
plt.plot(tt, ct, color='m', marker='^', label='TAU')
plt.plot(tp, cp, color='b', marker='s', label='PRO')
plt.legend(loc='bottom right')
plt.xlabel('Time (s)')
plt.ylabel('Concentration of absorbed CO$_{2}$ (M)')

plt.tight_layout()
for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/summaryplot' + ext, dpi=300)

plt.show()
```

5.3. Figure 5

```
import xlrd
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'CSTRLowLYS conc.')
sh2 = wb.sheet_by_name(u'CSTRLowMEA conc.')
```

```python

x = sh1.col_values(1,start_rowx=1) #Time
y = sh1.col_values(7,start_rowx=1) #lysine co2 loading
x1 = sh2.col_values(1,start_rowx=1)
y1 = sh2.col_values(6,start_rowx=1)#MEa co2 loading

import matplotlib.pyplot as plt
plt.figure(figsize=(3, 4))
plt.plot(x, y, label='K$^{+}$LYS$^{-}$')
plt.plot(x1, y1, label='MEA')
plt.legend(loc='best')
plt.xlabel('Reaction Time (min)')
plt.ylabel('CO$_{2}$ loading (mol CO$_{2}$/mol lysine/MEA)')

from matplotlib.ticker import MultipleLocator, FormatStrFormatter

majorLocator   = MultipleLocator(100)
majorFormatter = FormatStrFormatter('%d')

ax = plt.gca()
ax.xaxis.set_major_locator(majorLocator)
ax.xaxis.set_major_formatter(majorFormatter)

plt.tight_layout()
for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/lowcstrcompar' + ext, dpi=300)

plt.show()
```

5.4. Figure 6

```python
import xlrd
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'CSTR-HighLYS conc.')
sh2 = wb.sheet_by_name(u'CSTR-HighMEA conc.')

x = sh1.col_values(10,start_rowx=1)  # Time
y = sh1.col_values(7,start_rowx=1)  # lysine co2 loading
```

```
x1 = sh2.col_values(1,start_rowx=1)
y1 = sh2.col_values(7,start_rowx=1)  # MEA co2 loading

import matplotlib.pyplot as plt
plt.figure(figsize=(3, 4))
plt.plot(x, y, label='K$^{+}$LYS$^{-}$')
plt.plot(x1, y1, label='MEA')
plt.legend(loc='best')
plt.xlabel('t*(Q/V) (unitless)')
plt.ylabel('CO$_{2}$ loading (mmol CO$_{2}$/g solution)')

plt.tight_layout()
for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/highcstrcompar' + ext, dpi=300)

plt.show()
```

5.5. Figure 7

```
import xlrd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import OffsetImage, AnnotationBbox

wb  =  xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'MEARaman')
xf = sh1.col_values(0)
yf = sh1.col_values(2)  # Raman intensity postMF fingerprint region
xpf = sh1.col_values(3)
ypf = sh1.col_values(4)  # Raman intensity preMF fingerprint region
xh = sh1.col_values(6, end_rowx=2583)
yh = sh1.col_values(7, end_rowx=2583)  # Raman intensity postMF high region
xph = sh1.col_values(9, end_rowx=2583)
yph = sh1.col_values(10, end_rowx=2583)  # Raman intensity preMF high region

# data has been read in
fig = plt.figure()
```

```

# figure created
ax = fig.add_subplot(1, 2, 1)

# left subplot added
ax.plot(xpf, ypf, label='Pre-MF expt')  # plotting pre data

xpf = np.array(xpf)
ypf = np.array(ypf)
ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
shadeu = (xpf > 1400) & (xpf < 1600)
ax.set_ylim(20, 100)
ax.fill_between(xpf[shadeu], y1=np.zeros(len(xpf[shadeu])),
                y2=ymax*np.ones(len(xpf[shadeu])),
                color='#C8C8C8')
shadei= (xpf>800) & (xpf<1000)

ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
                y2=ymax * np.ones(len(xpf[shadei])), color='#B8B8B8')

ax.plot(xf, yf, label='Post-MF expt')  # plotting post data
ax.set_xlim([800, 1800])
ax.legend(loc='upper right')
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_ylabel('Raman Intensity (a.u.)')
ax.set_yticklabels([])

ax.text(1450, 30, r'CO$_{2}$$^{-}$')
ax.text(1420, 27, r'Stretch')
ax.text(1130, 76, r'1162')
ax.text(1145, 79, r'$\bigtriangleup$', color='r')
ax.text(1030, 92, r'1067')
ax.text(1045, 95, r'$\bigcirc$', color='r')
ax.text(910, 90, r'1017')
ax.text(950, 92.5, r'$\bigstar$', color='r')
ax.text(850, 35, r'CH$_{2}$')
ax.text(835, 32, r'Twist')
ax.text(840, 29, r'CCO')
```

```
ax.text(818, 26, r'Stretch')
ax.text(1020, 33, r'$\bigtriangleup$', color='r')
ax.text(1080, 33, r'Carbamate', color='r')
ax.text(1020, 30, r'$\bigstar$', color='r')
ax.text(1080, 29, r'HCO$_{3}$$^{-}$', color='r')
ax.text(1020, 26, r'$\bigcirc$', color='r')
ax.text(1080, 25, r'CO$_{3}$$^{-2}$', color='r')
# plot with shading
ax = fig.add_subplot(1, 2, 2)

# right subplot added
ax.plot(xph, yph, label='Pre-MF expt')
xph = np.array(xph)
yph = np.array(yph)
ymax = ax.get_ylim()[1]
shadex = (xph > 3200) & (xph < 3400)
ax.fill_between(xph[shadex],  y1=np.zeros(len(xph[shadex])),
                y2=ymax*np.ones(len(xph[shadex])),
                color='#D0D0D0')
shadeq = (xph>2800) & (xph< 3000)
ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
                y2=ymax*np.ones(len(xph[shadeq])),  color='#A8A8A8')
ax.plot(xh, yh,  label='Post-MF expt')
ax.set_xlim([2800, 3800])

ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_yticklabels([])

# plotting with shading
image = mpimg.imread('800px-Ethanolamine-2D-skeletal-B.png')

# read in image and make into array of colors
imagebox=OffsetImage(image,  zoom=0.15)
ax.text(3250,  300, r'NH$_{2}$')
ax.text(3250, 275, r'N-H')
ax.text(3210, 250, r'Stretch')
ax.text(3170, 570, r'3317')
ax.text(2850, 300, r'CH$_{2}$')
```

```python
ax.text(2850, 275,r'C-H')
ax.text(2810, 250, r'Stretch')

# rescaling size of image
structure = AnnotationBbox(imagebox, xy=(3300, 50), frameon=False)

# places image on plot at xy location
ax.add_artist(structure)
plt.tight_layout()

for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/MEAMFRamanwithstructure' + ext, dpi=300)

plt.show()
```

5.6. *Figure 8*

Raman spectra for GLY.

```python
import xlrd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import OffsetImage, AnnotationBbox

#reading in data
wb = xlrd.open_workbook('RawData.xlsx')
sh1=wb.sheet_by_name(u'GlyRaman')
xf=sh1.col_values(3) #wavenumber
yf=sh1.col_values(5) #Raman intensity postMF fingerprint region
xpf=sh1.col_values(0)
ypf=sh1.col_values(1)# Raman intensity preMF fingerprint region
xh=sh1.col_values(9, end_rowx=2583)
yh=sh1.col_values(10, end_rowx=2583)#Raman intensity postMF high region
xph=sh1.col_values(6, end_rowx=2583)
yph=sh1.col_values(7, end_rowx=2583)#Raman intensity preMF high region

```

```
# creating Figure
fig=plt.figure()

# adding left subplot
ax = fig.add_subplot(1, 2, 1)
ax.plot(xpf, ypf,  label='Pre-MF expt')  # plotting pre data

# making shading
xpf = np.array(xpf)
ypf = np.array(ypf)
shadeu = (xpf > 1400) & (xpf < 1600)
ax.set_ylim(5,  100)
ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
ax.fill_between(xpf[shadeu],  y1=np.zeros(len(xpf[shadeu])),
                y2=ymax*np.ones(len(xpf[shadeu])),
                color='#C8C8C8')
shadei= (xpf>800) & (xpf<1000)

ax.fill_between(xpf[shadei],  y1=np.zeros(len(xpf[shadei])),
                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
ax.plot(xf, yf,  label='Post-MF expt')#plotting post data
ax.set_xlim([800, 1800])
ax.legend(loc='upper right')

ax.text(1450, 14, r'CO$_{2}$$^{-}$')
ax.text(1410, 10, r'Stretch')
ax.text(1430, 57, r'1445')
ax.text(1135, 51, r'1175')
ax.text(1140, 54, r'$\bigtriangleup$', color='r')
ax.text(990, 71, r'1045')
ax.text(1010, 74, r'$\bigstar$', color='r')
ax.text(1000, 77, r'$\bigcirc$', color='r')
ax.text(900, 80, r'914')
ax.text(850, 22, r'CH$_{2}$')
ax.text(835, 19, r'Twist')
ax.text(850, 16, r'CCO')
ax.text(820, 13, r'Stretch')
ax.text(1500, 83, r'Carbamate', color='r')
```

```
ax.text(1430, 83, r'$\bigtriangleup$', color='r')
ax.text(1440, 79, r'$\bigstar$', color='r')
ax.text(1500, 79, r'HCO$_{3}$$^{-}$', color='r')
ax.text(1430, 75, r'$\bigcirc$', color='r')
ax.text(1500, 75, r'CO$_{3}$$^{-2}$', color='r')
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_ylabel('Raman Intensity (a.u.)')
ax.set_yticklabels([])

# adding subplot two
ax = fig.add_subplot(1, 2, 2)
ax.plot(xph, yph,  label='Pre-MF expt')

# making shading
xph = np.array(xph)
yph = np.array(yph)
ax.set_ylim(0, 700)
ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
shadex = (xph > 3200) & (xph < 3400)
ax.fill_between(xph[shadex],  y1=np.zeros(len(xph[shadex])),
                y2=ymax*np.ones(len(xph[shadex])),
                color='#D0D0D0')
shadeq = (xph>2800) & (xph< 3000)
ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
                y2=ymax*np.ones(len(xph[shadeq])),  color='#A8A8A8')
ax.plot(xh, yh,  label='Post-MF expt')
ax.set_xlim([2800, 3800])

ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_yticklabels([])

# text+image
ax.text(3250, 400, r'NH$_{2}$')
ax.text(3250, 370, r'N-H')
ax.text(3210, 345, r'Stretch')
ax.text(3245, 556, r'3320')
ax.text(2850, 400, r'CH$_{2}$')
ax.text(2850, 370, r'C-H')
```

```
ax.text(2810, 345, r'Stretch')
image = mpimg.imread('glycine.png')

# read in image and make into array of colors
imagebox=OffsetImage(image,  zoom=0.8)

# rescaling size of image
structure = AnnotationBbox(imagebox, xy=(3300,  100), frameon=False)

# places image on plot at xy location
ax.add_artist(structure)
plt.tight_layout()

for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/GLYMFRamanwithstructure.png', dpi=300)
plt.show()
```

5.7. Figure 9

```
import xlrd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import OffsetImage, AnnotationBbox

#reading in data
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'TauRaman')
xpf = sh1.col_values(0)  # wavenumber
ypf = sh1.col_values(1)  # Raman intensity preMF fingerprint region
xf = sh1.col_values(3)
yf = sh1.col_values(5)  # Raman intensity postMF fingerprint region
xph = sh1.col_values(6, end_rowx=2583)
yph = sh1.col_values(7, end_rowx=2583)  # Raman intensity preMF high region
xh = sh1.col_values(9, end_rowx=2583)
yh = sh1.col_values(10, end_rowx=2583)  # Raman intensity postMF high region

# Creating figure
```

```
fig = plt.figure()

# adding left subplot
ax = fig.add_subplot(1, 2, 1)
ax.plot(xpf, ypf,  label='Pre-MF expt')  # plotting pre data

# making shading
xpf = np.array(xpf)
ypf = np.array(ypf)
shadeu = (xpf > 1400) & (xpf < 1600)
ax.set_ylim(0,  400)
ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
ax.fill_between(xpf[shadeu],  y1=np.zeros(len(xpf[shadeu])),
                y2=ymax*np.ones(len(xpf[shadeu])),
                color='#C8C8C8')
shadei= (xpf>700) & (xpf<1000)

ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')
shadee= (xpf>1030) & (xpf<1070)
ax.fill_between(xpf[shadee], y1=np.zeros(len(xpf[shadee])), y2=ymax*np.ones(len(xpf[shadee])), color='#C0C0C0')

ax.plot(xf, yf,  label='Post-MF expt')#plotting post data
ax.set_xlim([700, 1700])
ax.legend(loc='upper right')
ax.text(1450, 100, r'CO$_{2}$$^{-}$')
ax.text(1410, 85, r'Stretch')
ax.text(1280, 60, r'1340')
ax.text(1310, 72, r'$\bigstar$', color='r')
ax.text(1020, 250, r'SO$_{3}$$^{-}$')
ax.text(1020, 230, r'Stretch')
ax.text(900, 80, r'955')
ax.text(800, 300, r'CH$_{2}$')
ax.text(800, 285, r'Twist')
ax.text(800, 270, r'CCO')
ax.text(800, 255, r'Stretch')
ax.text(1530, 300, r'HCO$_{3}$$^{-}$', color='r')
ax.text(1480, 300, r'$\bigstar$', color='r')
```

```
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_ylabel('Raman Intensity (a.u.)')
ax.set_yticklabels([])

# adding second subplot
ax = fig.add_subplot(1, 2, 2)
ax.plot(xph, yph,  label='Pre-MF expt')

# making shading
xph = np.array(xph)
yph = np.array(yph)
ax.set_ylim(0, 700)
ymax = ax.get_ylim()[1]+ax.get_ylim()[0]
shadex = (xph > 3200) & (xph < 3400)
ax.fill_between(xph[shadex],  y1=np.zeros(len(xph[shadex])),
                y2=ymax*np.ones(len(xph[shadex])),
                color='#D0D0D0')
shadeq = (xph>2800) & (xph< 3000)
ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
                y2=ymax*np.ones(len(xph[shadeq])),
                color='#A8A8A8')
ax.plot(xh, yh,  label='Post-MF expt')
ax.set_xlim([2800, 3800])

ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_yticklabels([])
#text+image
ax.text(3250, 300, r'NH$_{2}$')
ax.text(3250, 270, r'N-H')
ax.text(3210, 250, r'Stretch')
ax.text(3270, 560, r'3317')
ax.text(2850, 300, r'CH$_{2}$')
ax.text(2850, 270, r'C-H')
ax.text(2810, 250, r'Stretch')

image = mpimg.imread('Taurine.png')

# read in image and make into array of colors
```

```python
imagebox=OffsetImage(image,  zoom=0.4)

# rescaling size of image
structure = AnnotationBbox(imagebox, xy=(3300,  90), frameon=False)

# places image on plot at xy location
ax.add_artist(structure)
plt.tight_layout()

for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/TauMFRamanwithstructure' + ext, dpi=300)

plt.show()
```

5.8. Figure 10

```python
import xlrd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import OffsetImage, AnnotationBbox

# reading in data
wb = xlrd.open_workbook('RawData.xlsx')
sh1 = wb.sheet_by_name(u'ProRaman')
xpf = sh1.col_values(0)  # wavenumber
ypf = sh1.col_values(1)  # Raman intensity preMF fingerprint region
xf = sh1.col_values(3)
yf = sh1.col_values(5)  # Raman intensity postMF fingerprint region
xph = sh1.col_values(9, end_rowx=2583)
yph = sh1.col_values(10, end_rowx=2583)  # Raman intensity preMF high region
xh = sh1.col_values(6, end_rowx=2583)
yh = sh1.col_values(7, end_rowx=2583)  # Raman intensity postMF high region

# creating Figure
fig = plt.figure()

# adding left subplot
```

```
ax = fig.add_subplot(1, 2, 1)
ax.plot(xpf, ypf,  label=’Pre-MF expt’)  # plotting pre data

# making shading
xpf = np.array(xpf)
ypf = np.array(ypf)
shadeu = (xpf > 1400) & (xpf < 1600)
ax.set_ylim(0,  120)
ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
ax.fill_between(xpf[shadeu],  y1=np.zeros(len(xpf[shadeu])),
                y2=ymax*np.ones(len(xpf[shadeu])),
                color=’#C8C8C8’)
shadei= (xpf>800) & (xpf<1000)

ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
                y2=ymax*np.ones(len(xpf[shadei])), color=’#B8B8B8’)
shadeo= (xpf>1130) & (xpf<1190)
ax.fill_between(xpf[shadeo], y1=np.zeros(len(xpf[shadeo])),
                y2=ymax*np.ones(len(xpf[shadeo])), color=’#B0B0B0’)

ax.plot(xf, yf,  label=’Post-MF expt’) # plotting post data
ax.set_xlim([800, 1800])
ax.legend(loc=’upper right’)
ax.text(1450, 16, r’CO$_{2}$$^{-}$’)
ax.text(1410, 14, r’Stretch’)
ax.text(1410, 8, r’R$_{1}$R$_{2}$NH’)
ax.text(1450, 4, r’N-H’)
ax.text(1410, 0.5, r’Stretch’)
ax.text(1275, 75, r’1350’)
ax.text(1310, 79, r’$\bigstar$’, color=’r’)
ax.text(1070, 14, r’R$_{1}$R$_{2}$NH’)
ax.text(1093, 10, r’C-N-C’)
ax.text(1080, 6, r’Stretch’)
ax.text(850, 20, r’CH$_{2}$’)
ax.text(840, 16, r’Twist’)
ax.text(845, 11, r’CCO’)
ax.text(820, 7, r’Stretch’)
ax.text(1630, 95, r’HCO$_{3}$$^{-}$’, color=’r’)
```

```
ax.text(1592, 96, r'$\bigstar$', color='r')
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_ylabel('Raman Intensity (a.u.)')
ax.set_yticklabels([])

# adding subplot two
ax = fig.add_subplot(1, 2, 2)
ax.plot(xph, yph,  label='Pre-MF expt')

# making shading
xph = np.array(xph)
yph = np.array(yph)
ax.set_ylim(0, 700)
ymax = ax.get_ylim()[1]+ax.get_ylim()[0]

shadeq = (xph > 2800) & (xph < 3000)
ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
                y2=ymax*np.ones(len(xph[shadeq])),
                color='#A8A8A8')
ax.plot(xh, yh,  label='Post-MF expt')
ax.set_xlim([2800, 3800])
ax.text(2850, 400, r'CH$_{2}$')
ax.text(2850, 370, r'C-H')
ax.text(2810, 350, r'Stretch')
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_yticklabels([])

# text + image
image = mpimg.imread('Proline.png')

# read in image and make into array of colors
imagebox=OffsetImage(image,  zoom=0.8)

# rescaling size of image
structure = AnnotationBbox(imagebox, xy=(3300,  100), frameon=False)

# places image on plot at xy location
ax.add_artist(structure)
```

```
plt.tight_layout()

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

5.9. Figure 11

```
import xlrd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import OffsetImage, AnnotationBbox

# reading in data
wb = xlrd.open_workbook('RawData.xlsx')
sh1=wb.sheet_by_name(u'LysRaman')
xpf=sh1.col_values(0)  # wavenumber
ypf=sh1.col_values(1)  # Raman intensity preMF fingerprint region
xf=sh1.col_values(3)
yf=sh1.col_values(5)  # Raman intensity postMF fingerprint region
xph=sh1.col_values(9, end_rowx=2583)
yph=sh1.col_values(10, end_rowx=2583)  # Raman intensity preMF high region
xh=sh1.col_values(6, end_rowx=2583)
yh=sh1.col_values(7, end_rowx=2583)  # Raman intensity postMF high region

# creating Figure
fig=plt.figure()

# adding left subplot
ax = fig.add_subplot(1, 2, 1)
ax.plot(xpf, ypf,  label='Pre-MF expt')  # plotting pre data

# making shading
xpf = np.array(xpf)
ypf = np.array(ypf)
shadeu = (xpf > 1400) & (xpf < 1600)
ax.set_ylim(10,  120)
```

```
ymax = ax.get_ylim()[1] + ax.get_ylim()[0]
ax.fill_between(xpf[shadeu],  y1=np.zeros(len(xpf[shadeu])),
                y2=ymax*np.ones(len(xpf[shadeu])),
                color='#C8C8C8')
shadei= (xpf>800) & (xpf<1000)

ax.fill_between(xpf[shadei], y1=np.zeros(len(xpf[shadei])),
                y2=ymax*np.ones(len(xpf[shadei])), color='#B8B8B8')

ax.plot(xf, yf,  label='Post-MF expt')  # plotting post data
ax.set_xlim([800, 1800])
ax.legend(loc='upper right')
ax.text(1450, 25, r'CO$_{2}$$^{-}$')
ax.text(1410, 21, r'Stretch')
ax.text(1100, 80, r'1130')
ax.text(1100, 83, r'$\bigtriangleup$', color='r')
ax.text(1030, 106, r'1068')
ax.text(1030, 110, r'$\bigcirc$', color='r')
ax.text(930, 96.5, r'1017')
ax.text(992, 100, r'$\bigstar$', color='r')
ax.text(850, 40, r'CH$_{2}$')
ax.text(830, 36, r'Twist')
ax.text(850, 28, r'CCO')
ax.text(820, 24, r'Stretch')
ax.text(1510, 100, r'Carbamate', color='r')
ax.text(1450, 100, r'$\bigtriangleup$', color='r')
ax.text(1460, 95, r'$\bigstar$', color='r')
ax.text(1510, 95, r'HCO$_{3}$$^{-}$', color='r')
ax.text(1450, 90, r'$\bigcirc$', color='r')
ax.text(1510, 90, r'CO$_{3}$$^{-2}$', color='r')
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_ylabel('Raman Intensity (a.u.)')
ax.set_yticklabels([])

# adding subplot two
ax = fig.add_subplot(1, 2, 2)
ax.plot(xph, yph,  label='Pre-MF expt')

```

```
# making shading
xph = np.array(xph)
yph = np.array(yph)
ax.set_ylim(0, 800)
ymax = ax.get_ylim()[1]+ax.get_ylim()[0]

shadeq = (xph>2800) & (xph< 3000)
ax.fill_between(xph[shadeq], y1=np.zeros(len(xph[shadeq])),
                y2=ymax*np.ones(len(xph[shadeq])),  color='#A8A8A8')
shadey= (xph>3200) &(xph<3400)
ax.fill_between(xph[shadey], y1=np.zeros(len(xph[shadey])),
                y2=ymax*np.ones(len(xph[shadey])),  color= '#A0A0A0')

ax.plot(xh, yh,  label='Post-MF expt')
ax.set_xlim([2800, 3800])
ax.set_xlabel('Wavenumber (cm$^{-1}$)')
ax.set_yticklabels([])

# text+image
ax.text(3250, 400, r'NH$_{2}$')
ax.text(3250, 370, r'N-H')
ax.text(3210, 350, r'Stretch')
ax.text(3250, 730, r'3310')
ax.text(2850, 400, r'CH$_{2}$')
ax.text(2850, 370, r'C-H')
ax.text(2810, 350, r'Stretch')
image = mpimg.imread('Lysine.png')

# read in image and make into array of colors
imagebox=OffsetImage(image,  zoom=0.65)

# rescaling size of image
structure = AnnotationBbox(imagebox, xy=(3320,  75), frameon=False)

# places image on plot at xy location
ax.add_artist(structure)
plt.tight_layout()

```

```python
for ext in ['.png', '.eps', '.pdf']:
    plt.savefig('../figures/LYSMFRamanwithstructure' + ext, dpi=300)

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.