**Installation  
(Windows)**

If you wish to interact with the provided computational code directly, the quickest method for starting out with Python is to download the software through the Anaconda platform. There are multiple methods for starting out with python, but for consistency, we recommend Anaconda.

1. To download Anaconda for Windows, navigate to the following webpage: https://www.anaconda.com/distribution
2. In the tab labeled “Anaconda for Windows”, there are installers for Python 2.7 and Python 3.7. The provided program was written in Python 3.7.3. While there is some utility in installing both versions of Python for personal use, or the purposes of this program, 3.7 is recommended.
3. In the box labeled “Python 3.7”, click on the link labeled “64-bit graphical installer.” If you are using Windows 32-bit, the graphical installer is provided below the 64-bit installer.
4. Download of the ‘.exe’ file will commence upon clicking the link. No more interaction with the anaconda website is necessary.
5. Navigate in your file explorer to the previously downloaded ‘.exe’ file, and open the file.
6. The Anaconda3 Setup program will initialize. On the initial window, click ‘Next’
7. “Read” the License Agreement and click ‘I Agree’
8. Under the next window, **Select Installation Type**, the option ‘Just Me (recommended)’ should be toggled. Click ‘Next >’
9. Under the next window, **Choose Install Location**, the ‘Destination Folder’ should be displayed (default is sufficient). Click ‘Next >’
10. Under the next window, **Advanced Installation Options**, the option ‘Register Anaconda as my default Python 3.7’ should be check marked. Click ‘Install’
11. Installation should commence. Such should take a few minutes.
12. Upon installation, A dialogue box for ‘Microsoft VSCode’ may pop up. Click ‘Skip’. On the next window, for the purposes of this specific installation, uncheck both boxes before clicking ‘Finish’ (though you’re certainly welcome to review the information provided). From here, several executable programs are now available on your computer. The recommended program for use is an interactive development environment called ‘Spyder’. Pressing the windows button and typing in the search bar ‘Spyder’ should yield an access point to the IDE. It’s recommended that a shortcut be generated on the desktop to this environment. Alternatively, search for ‘anaconda prompt’ and when the terminal opens, type ‘spyder’ and hit enter.

**Installation   
(Linux Ubuntu/LinuxMint)**

If you wish to interact with the provided computational code directly, the quickest method for starting out with Python is to download the software through the Anaconda platform. There are multiple methods for starting out with python, but for consistency, we recommend Anaconda.

1. To download Anaconda for Windows, navigate to the following webpage: https://www.anaconda.com/distribution
2. In the tab labeled “Anaconda for Linux”, there are installers for Python 2.7 and Python 3.7. The provided program was written in Python 3.7.3. While there is some utility in installing both versions of Python for personal use, or the purposes of this program, 3.7 is recommended.
3. In the box labeled “Python 3.7”, click on the link labeled “64-bit (x86) installer.”
4. Open the Linux Terminal and type some form of the following:

bash /file\_location/file\_name.sh

the file location and file name can be found by right-clicking the downloaded ‘.sh’ file and selecting ‘properties’.

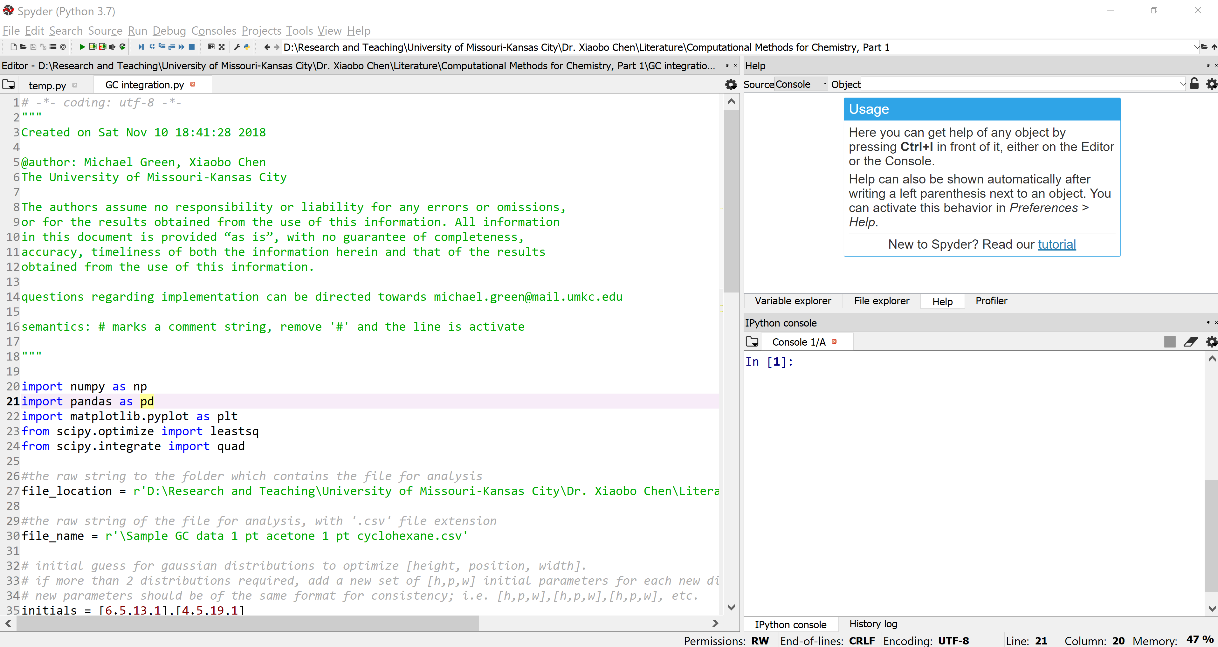
1. A welcome prompt will appear. Press ‘Enter’. (you can hold the enter key to scroll through the EULA until you reach the next input prompt)
2. Type ‘yes’ if you accept the EULA.
3. Once Anaconda is installed, type into the Linux terminal ‘spyder’ and press enter.

**Installation  
(macOS)**

If you wish to interact with the provided computational code directly, the quickest method for starting out with Python is to download the software through the Anaconda platform. There are multiple methods for starting out with python, but for consistency, we recommend Anaconda.

1. To download Anaconda for macOS, navigate to the following webpage: https://www.anaconda.com/distribution
2. In the tab labeled “Anaconda for macOS”, there are installers for Python 2.7 and Python 3.7. The provided program was written in Python 3.7.3.
3. Pressing the ‘download’ button will toggle a save request by the mac. Go ahead and save the distribution to the recommended file location. Once downloaded, launch the package installer.
4. Hit continue until you reach the EULA, feel free to glance through its contents before hitting ‘agree’.
5. The next page will allow the user to specify the installation folder. Feel free to select the default setting.
6. When all of the necessary installation steps are completed, click ‘install’.
7. Once installation is complete, feel free to move the anaconda installer to the trash.
8. In the macOS launchpad, open the macOS terminal. Type ‘spyder’ and press enter.

**Operation Instructions**

1. Once Spyder is open, navigate to the top left of the main window, and click ‘File’. In the dropdown menu, click ‘open’ and navigate to the ‘GC integration.py’ file included in the supporting information, select the file, and click ‘open’. If this proves difficult, the .py file can also be opened in notepad to be copy/pasted into the text window.
2. A few things to note: in the left window is your code. If you’re starting from scratch, the most important thing to know is that Python is what’s called ‘object oriented’. I.E. you work with ‘objects’ by defining them. Objects include variables (like x = 1, the variable x is equivalent to 1), lists (which is a list of objects, like numbers or variables), arrays (like a 2d list), and strings (variable = ‘this is a string’), among others. It’s recommended to try out defining some variables for fun over in the ‘IPython console’ in the bottom right quadrant of the Spyder window. Click to the right of the In [1]:text to activate the window and try typing in the following definitions:
   1. x0 = 1
   2. x0 = x0 + 1
   3. x1 = [1,2,3]
   4. x1[1] = 4
   5. words = ‘this is a string’

after tying each of the expressions, type in the variable you just defined and hit the ‘enter’ key. The output should give you what you just defined the object to be. To note, for option b, you’re redefining a previously defined object. Furthermore, for option d, the console is taking your previous definition of the object ‘x1’ and redefining the ‘1’ position of the list to be 4. So, for c where x1 = [1,2,3], the operation executed by d. will redefine the [1] position to be ‘4’. Lists and Array indexes start at 0.

1. Included in the supplemental are two ‘.csv’ files of what contain X,Y data from GC analysis. Save those files somewhere on your computer – for example, the desktop. The location of those files should be placed as a string inside the variable ‘file\_string’ in the program, between the two quotation marks (note, for Windows users, python requires an extra “r” mark at the beginning of the string, because the forward-slash is a special lexicon character. We avoid the issue by designating the string as “raw” with an “r”).   
     
   So as example, the object should read something like:  
     
   **Windows:** file\_location = r'C:\Users\1mike\Desktop\Sample GC 1 pt acetone 1 pt cyclohexane.csv'  
   **Linux:** file\_location = ‘/home/michael/Desktop/Sample GC 1 pt acetone 1 pt cyclohexane.csv’  
   **macOS:** file\_location = ‘/Users/1mikegrn/Desktop/Sample GC 1 pt acetone 1 pt cyclohexane.csv’

If you’ve renamed the file, make sure the ‘file\_name’ variable reflects the accurate path accordingly.

1. Once the file string is updated, press ‘f5’ to run the program in the IPython console. Alternatively, press the green play button at the top left of the window. The output is a graph of the data, with the respective curves, functions, and value of the fit. You can also press ‘f10’ to run the code in a different terminal.

The way the program works in layman terms is that, given an initial set of parameters, the least-squares regression analysis will iteratively correct the initial parameters in a manner that minimizes the error between the function and the data. As such, the initial parameters should be close to the true result, but can be off. At minimum, try to guess the height and position parameters of the gaussian functions (initials = [h,p,w] with h and p being height and position respectively). If your initial guesses are extremely off, (for example, initials = [0,0,0],[100,100,100]), the computer will minimize to a local minimum, but the local minimum will not be the global minimum. If you’re having issues after trying to run the program and want to look at the data directly, copy and paste into the IPython window, ‘plt.plot(M[:,0],M[:,1])’ and be sure the initial parameters are near the top of the peaks. If an error is thrown, it means that your IDE isn’t finding the data.

A good, cheap reference for further reading into general use of Python is *Learning Scientific Programming with Python* by Christian Hill.

Finally, if the data is to be read from an excel file, you can change the ‘master’ variable to read:  
master = pd.read\_excel(file\_string)

More on the specifics of this program and the utility is provided as comments in the ‘.py’ script.