Experiment 7a: Window Analysis for Optimizing Isocratic HPLC Separations.

A Python implementation of an LC simulation exercise by Gary Mabbott, St. Thomas University (St. Paul, MN).

Prelab Assignment

1. Look up the definition and write out the meaning and/or equation for the following terms (use Harris' Table 23-2):
    a. Chromatogram:
    b. Stationary phase:
    c. Mobile phase:
    d. Eluent:
    e. Elution:
    f. Flow rate:
    g. Retention time:
    h. Adjusted retention time:
    i. Retention factor:
    j. Relative retention:
2. Solvent passes through a column in 3.0 minutes, but an analyte requires 9.0 minutes.
    a. Calculate the retention factor, k.
    b. What fraction of time is the analyte in the mobile phase as it passes through the column?
3. Complete the following questions from 9th ed. Harris: 23-C, and 23-17 (10th ed. 23-C, 23-18)
4. Prepare your ELN Exp 7a Data & Obs section.

Lab Activities

Simulator Setup. Go to the Multi-Dimensional Separations website (multidlc.org), select Tools, and then 1-D LC Simulator (or go directly to this link). Select "Manage Compounds" and use the checkboxes so that only *N*-benzylformamide, benzylalcohol, phenol, diethylformamide, methylparaben, and acetanilide are selected. Click on Custom Compound and add Caffeine, using the following parameters:

	Acetonitrile	Methanol
ln k_w intercept	1.5608	2.0909
ln k_w slope	-0.0150	-0.0217
S intercept	-9.056	-6.2387
S slope	0.0699	0.0400

Note that k_w is the estimated retention factor of the analyte (caffeine) eluting in pure water, S is solvent strength towards the analyte, and the intercept and slope describe the variation of these two parameters with temperature. Also note that if you leave the simulator by clicking on any of the menu items on this page, you will need to re-enter these 8 parameters! Once you have the seven compounds selected, including caffeine, click Manage Compounds again to close that area.

Part 1: Determining the relationship between the retention factor and mobile phase composition

Select Mobile Phase Composition, set Solvent B to Acetonitrile, and select “Isocratic Elution Mode.” This means the blend of solvent A (water) and B (acetonitrile) will stay constant throughout the run. Adjust the mobile phase composition using the slider bar until you get baseline resolution between all 7 compounds in your mixture. That means that you can see a little bit of the baseline between each pair of peaks. If you don’t see 7 peaks, that means some of your analytes are not yet separated! *In your ELN, note the fraction, ø, of the organic solvent that corresponds to your optimal chromatogram. Also note the retention times and identities of each peak under these conditions.*

Unfortunately, finding the optimum conditions for separating a real mixture takes much more time (and solvent) in the lab. Every adjustment you made to the mobile phase just now would have been a 20-30 minute experiment; plus, after optimizing the separation you’d have needed to inject your compounds one by one to determine which was which! However, we could save time and effort in the lab if we were able to use a small number of experiments to predict *retention factors* for each compound under a wide range of different solvent conditions. We will therefore use the simulator to gather some data about this.

We are going to mathematically model the retention as a function of solvent composition. For now, focus on **caffeine** and **diethylformamide**. *In your ELN, create a table for the retention factor, k, for both compounds at each solvent composition, ø, from 10% CAN (ø = 0.10) to 80% ACN in steps of 10%.* (In the table enter ø in decimal form, that is, ø = (%v/v organic modifier)/100. Use Python to create graphs of *k* vs. ø for both compounds. *Which of the following mathematical models best describes the relationship between k and ø?* (Fit the data for both compounds all 3 ways before deciding.) *Upload sample graphs and your best mathematical model to the ELN, including values of the constants m and b (or z2, z1, and z0)*

a) Linear: $k = b + m\o$
b) 2nd order polynomial: $k = (z2)\o^2 + (z1)\o + (z0)c$
c) Exponential: $\ln k = b + m\o$

Part 2: Applying the model

Return to the HPLC simulator and find the corresponding values for *k* for each of the five other compounds for 20 % organic modifier and 80 % organic modifier. Once again, use Python to estimate the parameters (*m*, *b*, *or z2, z1, and z0*) for the model that you have decided best relates *k* to ø. (Notice that once you know the general form of the model, this data can be estimated from results from just two chromatograms.)

The mathematical model will let us predict the behavior of these compounds at other solvent conditions. This could save time in running real separations of this mixture in the lab. Let’s assume that we want to resolve all of the components in this mixture isocratically. Use the coefficients for the equations that you determined above, and “while” loops in Python, to calculate estimates for ln*k* for each of these compounds for ø from 0.1 to 0.8 in steps of 0.01, storing them in a large 2-D array containing almost 500 ln*k* values.

Next, at each value of ø calculate ABS(lnk_2-lnk_1) for the closest pair of values for ln$k$. (Notice that this is equivalent to calculating the log of the separation factor, $\ln(\alpha) = \ln(k_1/k_2)$ for the least well-separated peaks at the corresponding value of *ø*.) For convenience let's call this term the selectivity. To do this on our large dataset without making us work too hard, we need to break it down into a sequence of small, simple operations that Python can do. (Breaking down complex tasks into simple steps is the essence of coding!) We will first use a *sort* function nested in a while loop to sort all of our peaks at each value of ø. Then, we can have Python calculate the difference between each adjacent peak on each chromatogram, and finally determine the closest pair of peaks at each value of ø, which we can call "minimum selectivity."

We'll then graph this minimum selectivity vs ø (connecting the points with a line). This process produces a diagram called a **window plot**. The top of each peak in a window plot represents a solvent mixture which gives a (local) maximum of peak separation. The apex of the tallest peak in the plot indicates conditions where the resolution is likely to be the best. Build this in Python, and then answer the following questions:

1. Do the conditions that you noted in the very first part of this exercise coincide with the top of one of the peaks in your window plot?

2. Are there other peaks that predict good resolution, or even better resolution that the conditions you originally selected? If so, run the HPLC simulator again under those conditions. Are all of the peaks well resolved? Why / why not?

3. If more than one set of conditions will produce a chromatogram with baseline resolution for all components in your mixture, what other considerations should be used in deciding the optimum operating conditions? This analysis considers only the selectivity factor. What other characteristics affect the resolution?

To complete the Results and Analysis section,

- Make sure that all your graphs and tables from part 7a have narrative (a few written sentences guiding a reader through each one – what are the important take-aways to notice?).
- Graphs should also have captions that say what they are and identify the meanings of colors and symbols.
- Tables should also have descriptive titles and column headings, with any units in the column headings.
- Be sure include a few sentences of conclusion.