Data Plotting and Curve Fitting with SciDAVis

David P. Goldenberg

August 12, 2022

This tutorial was originally written for a biochemistry laboratory class, Biol 3515/Chem 3515 at the University of Utah, http://goldenberg.biology.utah.edu/courses/biol3515/index.shtml. It is centered around data from a particular experiment, the measurement of protein concentrations using the Bradford Dye binding assay. However, this example should offer a useful guide for a variety of other applications.

SciDAVis an open-source program for graphing and analyzing numeric data. It has capabilities similar to a number of commercial programs, such as Kaleidagraph, SigmaPlot, Origin and IgorPro. The design of SciDAVis is similar to that of Origin, and SciDAVis can import at least some Origin files.

SciDAVis is available for Macintosh, Windows and Linux platforms. Files generated on a Macintosh can be read by the Windows version of SciDAVis, provided that they are saved on a Windows formatted disk or transferred over a network. The program has been installed on the computers in the Biology Department computer lab (106 S. Biol.).

SciDAVis can be downloaded from:
https://sourceforge.net/projects/scidavis/files/SciDAVis/
Different folders on this page contain different versions of the program, with the newest at the top. SciDAVis is frequently updated, which is generally a good thing, but sometimes new bugs are introduced that may take a while to be identified and corrected. As of August 2022, version 2.8 was the most recent, and this version appears to work very well.

Each version folder has three files in it. The files for Macs have a name with the suffix ".dmg". The file name for the Windows version has the suffix ".msi". The third file, with suffix ".gz" is a compressed archive file containing the source code for the program, as opposed to the executable files for specific platforms. Unless you really know what you are doing, and want to spend a lot of time on it, you probably don't want to download this file!

Executable versions for various flavors of Linux are available at:
http://software.opensuse.org/download.html?project=home%3Ahpcoder1&package=scidavis

Some cautions

Although SciDAVis is a very capable program, it lacks some of the polish of commercial programs and is far from bug free. The latest versions are much improved, however, and you should be able to use the without too much trouble once you learn its quirks. Here are two things to be aware of from the start:

1. Opening SciDAVis files. Unlike most files and programs on a Macintosh, double-clicking a SciDAVis file icon does not open the file in the program. If SciDAVis is not already open, double clicking on a project file icon will open the program, but not the project file. Instead, project files must be opened using the **Open** command in the **File** menu.

2. Applying changes to data tables and graphs. Many of the dialog boxes used to modify the properties of data tables and graphs include an **Apply** button that must be clicked befor the changes become effective. It is easy to forget to do this!

A.   Entering and manipulating data

1. Start the SciDavis program by "double clicking" on the program icon:

After the program starts, a large window should appear:
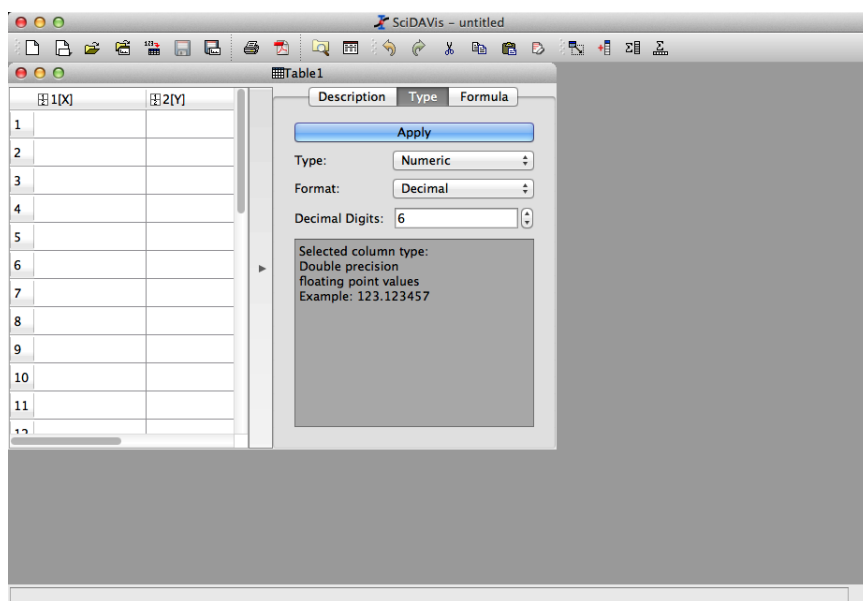
Files saved by SciDAVis represent "projects" and all of the data tables and graphs associated with a project are displayed, along with other information, in a single window like the one shown above.

2. When a new project window appears, it should contain a sub-window for a data table, as shown in the left portion of the window shown above. Initially the table will only contain two columns for data, but more columns can be added.

Begin by entering the data for your Bradford assay calibration curve, using the first column for the volumes of BSA solution used for each tube and the second column for the absorbance measured for each. You can move the cursor among the data cells using the mouse, the arrow keys or the return key, which will move the cursor down one cell. The data window should now look something like this:

The first column contains the number of microliters of BSA solution used in each assay and the second contains the absorbance measurements.

Rather than typing in the individual values, you can copy your data from LabArchives and paste it into the data window. If you do this, however, be sure to copy only the numeric values, and not any of the text headers from LabArchives. Including the text will likely cause SciDAVis to recognize all of the data in a column as text, rather than numeric data, and you will have trouble later when you try plotting the data.

3. This is a good time to save your project file. This is done using the **Save Project As...** command in the **File** menu. On the computers in the lab, it may be most convenient to save the file to the Desktop. When you are done, be sure to copy the files created in this session to your LabArchives notebook.

4. Next, rename the column names from "1" and "2" to something more meaningful. Next to the data columns themselves, there should be panel in which various properties of each column can be modified. If this panel is not visible, click on the small arrowhead on the right-hand side of the window. If opening this panel hides the data columns, the window can be stretched by clicking on the right-hand border and dragging.

   If the button labeled **Description** isn't already highlighted, click on it, and click on the top of the first column. You can now type a new name in the field labeled **Name**. For the first column, use the name "μL BSA".[1] To apply the new name, you must click the **Apply** button. (This is a general quirk of SciDAVis: almost all actions require clicking an **Apply** button.) For now, you can ignore the "[X]" that also appears in the column head.

   Similarly, label the second column A595.

   Clicking on the **Type** button will create another dialog for changing other properties of the data column, for instance allowing the use of text or dates or specifying the number of digits displayed. The third button, labeled **Formula** is used to create data from a user-specified formula.

---

[1]On a Mac, you can enter the μ symbol by holding down the option key and typing "m". Unfortunately, this doesn't work on Windows, so that you will need to settle for using "u"

5. At this point we could make a plot of absorbance versus BSA volume, but what we really want is to plot absorbance versus μg of BSA. This is a good opportunity to use SciDAVis to calculate the contents of new data columns.

First, create a new column to hold the μg data. From the **Table** menu, execute the **Add Column** command:
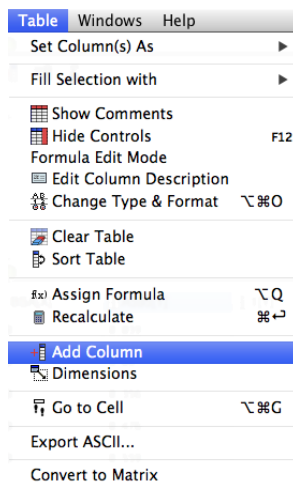
This should create a new column to the right of the existing ones. Use the Description dialog box to label this column "μg BSA", as you did earlier.

6. Select the new column and click on the **Formula** button. The new dialog box will enable you to calculate the contents of the new column in terms of the contents of other columns. The box should initially look like this:
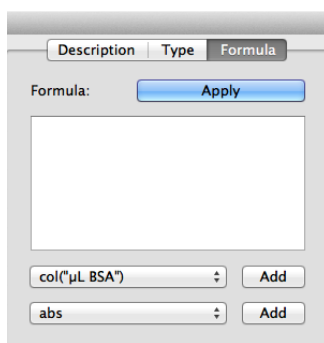
In our case, we want to calculate the number of micrograms of BSA in each sample. Since the stock solution contained 0.1 μg/μL, the number of micrograms is simply 0.1 times the number of microliters.

Note the set of four buttons at the bottom. These allow you to add terms to the formula. The upper two buttons are for adding references to other columns in the table. Clicking on the upper left button will create a drop-down menu with the names of the existing columns. Use this button to select the item col("μL BSA"). Then

4

click the upper **Add** button to add this term to the formula. To multiply this term, type "0.1 *" to the left of the term col("μL BSA"), so that the formula is:
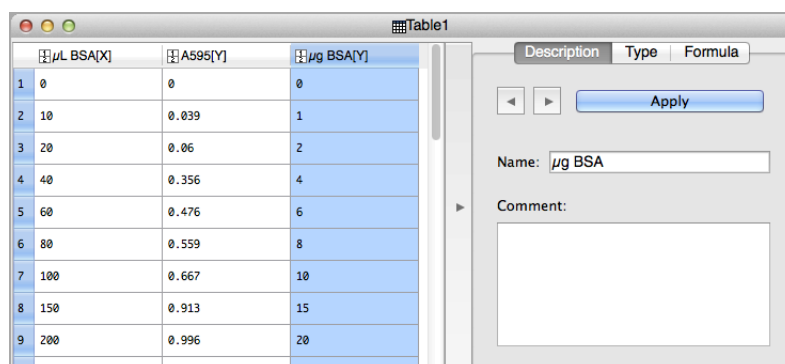
0.1*col("μL BSA")

In SciDAVis and most other computer programs, multiplication is indicated by the asterisk.

Then click the **Apply** button. The new column should now be filled with the results of multiplying the column "μL BSA" by 0.1. Notice that the rows below those containing any values for "μL BSA" now contain zeros. This isn't really a problem, but if you want to clear these cells, simply select them with the mouse and execute the **Delete Selection** command in the **Edit** menu. Be sure not to delete anything else!

This is obviously a very simple example, but the Formula box can be used to carry out very complex calculations as well. A wide variety of mathematical functions are built into the program and can be accessed by clicking on the lower left button in the Formula box.

7. At this point, your data window should look like:

Be sure to save your data before proceeding.

B. Plotting data

Now, it is time to make a graph! There are two ways (at least) in SciDAVis to generate a plot from selected columns of data, described below.
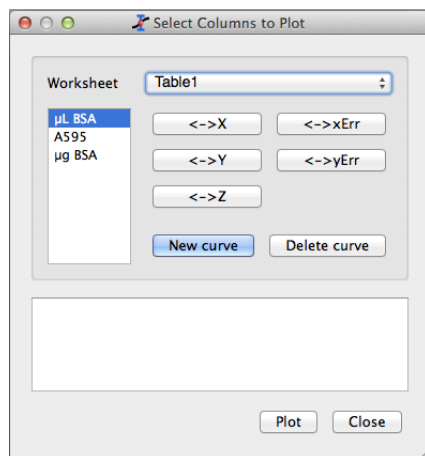
The Plot Wizard

The first way to create a plot is to use a dialog box, called the **Plot Wizard** to select specific data columns to represent on the X- and Y-axes.

1. First, select **Plot Wizard** from the **View** menu.
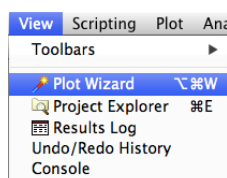
2. This will open a dialog box:

This dialog allows you to select any of the columns in a table to use as the X-(horizontal) or Y- (vertical) variables in a plot, and make multiple plots on the same graph. The individual pairs of X- and Y-variables are referred to as "curves".
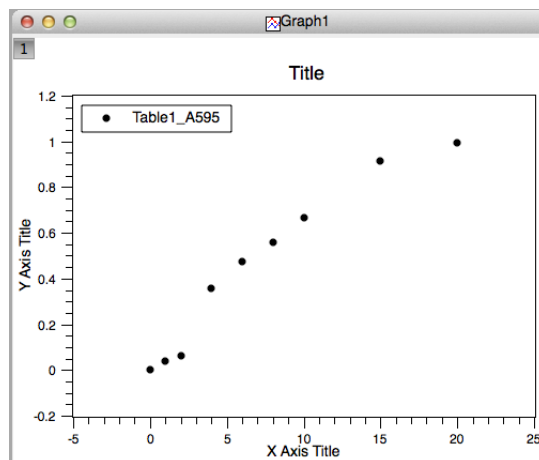
First, click on the **New Curve** button. Then select "μg BSA" from the column names and click on the button labeled **<->X** to make this column the X-variable.

Then, select, "A595" from the column names and click on the button labeled **<->Y**.

The dialog box should now look like:
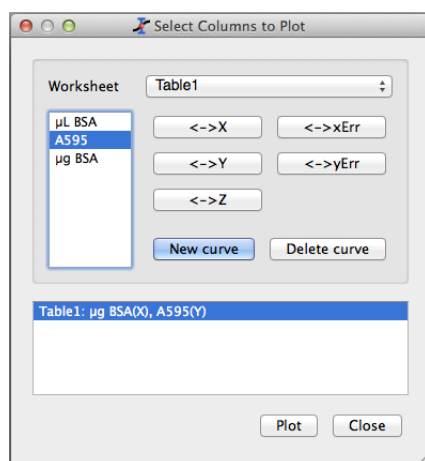
3. Click on the **Plot** button. You should now have plot that looks like:

Some of the details of the plot, such as the size and shapes of the symbols or whether line segments are drawn between the symbols, will depend on preferences that have been set for the program.

Before proceeding further, be sure to save the project.

Plotting from selected columns in a Data Table

Data columns in a table can also be selected directly with the mouse and then plotted by using commands in the **Plot** menu. This requires a bit of care and planning, however, as detailed below.

Columns in a data table are selected by clicking on the header cells (containing the column names) at the top of the window. The columns can be selected individually, or as groups of adjacent columns, as shown below for the columns named "μLBSA" and "A595". To select multiple adjacent columns, hold down the shift key while clicking on the headers of the individual columns.

Note that the two columns are also labeled X and Y, which defines how they will be treated by commands in the **Plot...** menu. For instance, if the **Scatter** command in the **Plot** menu is used with the selection shown above, a plot of A_{595} (on the Y-axis) versus μL BSA (on the X-axis) will be created.

In the window shown above, the column labeled "μg BSA" is identified as Y and must be modified to be used as an X-variable. To do this, select just this column, and select **Set Column(s) As** from the **Table** menu, and select **X** from the submenu that appears:
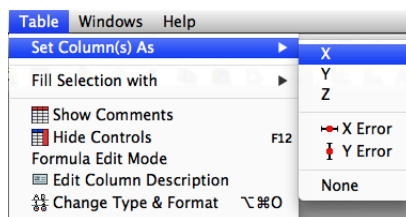
The headings on the columns will then be changed as shown below:

Notice that the μg BSA column is now labeled X2, and the "μL BSA" and "A595" columns are now labeled X1 and Y1. Although multiple columns in a table can be identified as Y, different X columns must be distinguished. (The exact numbers associated with different X and Y columns may differ, depending on the order in which various assignments were made.)

Another requirement for making plots directly from a data window is that the X- and Y-columns must be placed adjacent to one another. The positions in a table can be easily changed by using the mouse to click and drag on the column header cells. To make a plot of A_{595} versus μg BSA, the columns should be rearranged and selected as shown below:


Now, the two columns are identified as X2 and Y2 and will be properly plotted by using the commands in the **Plot** menu.

C. Modifying the appearance of the graph

Depending on how the preferences for the program are set, a plot generated from the **Plot Wizard** dialog will include symbols representing the data points, line segments connecting the data points or both. A plot containing only symbols is called a scatter plot in SciDAVis, a plot containing only line segments is called a line plot, and a plot containing both is called a symbol+line plot. But, there are many options available to modify the appearance of the plot after it is created.

1. To begin modifying the graph, first make sure that the graph window that you want to work on is active (selected), and then select **Plot...** from the **Format** menu. This will open a new dialog box, as shown below:
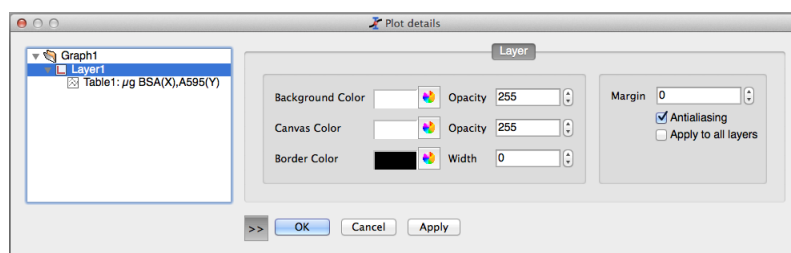
The white area on the left of this box indicates the hierarchy of the components making up a plot window, which can contain multiple layers, each of which can contain multiple curves, as defined in the **Plot Wizard** window. Our graph is relatively simple, with just one layer and one curve. The layer properties include things like the background and border colors, and can usually be left as they are.

Clicking on the icon representing the curve, changes the dialog box so that it appears like:
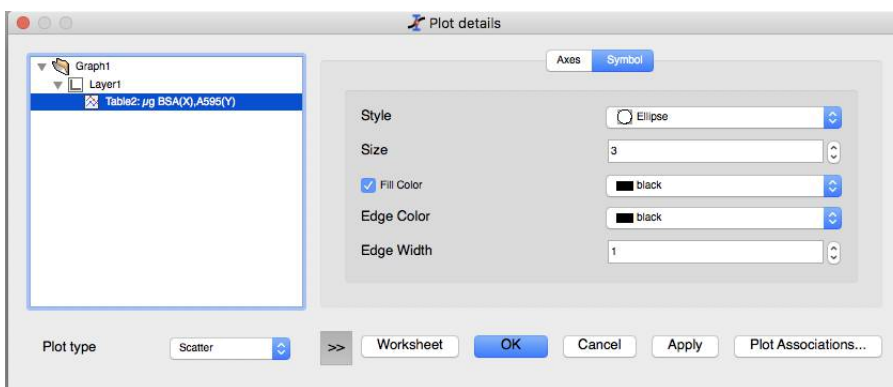


The buttons at the top in the window shown above, labeled **Axes** and **Symbol** allow different properties of the curve to be changed. The particular buttons that appear depend on the the type of plot, as described below.

With the **Symbol** button clicked, the window shows options for the plot symbols, including their shape, size and colors. You can play with these options to see their

9

effects. For some (but not all) of them, you will need to click the **Apply** button in order to see their effects.

If you chose the option of No Symbol, the plot style is converted to Line, and the dialog box changes to present options for the way in which the line segments are drawn, as shown below:
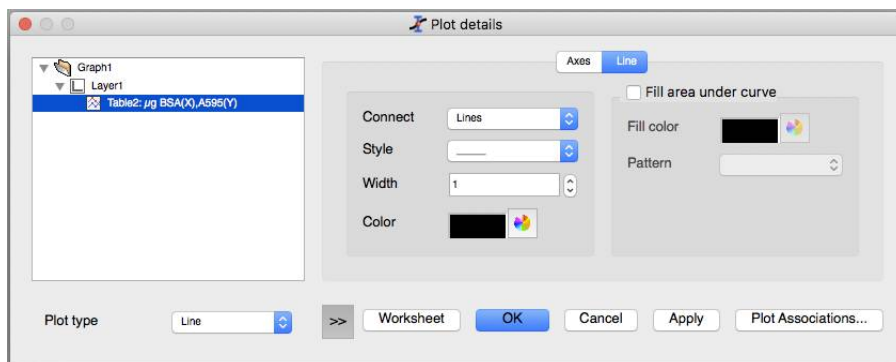
Among the options for the line style is "No line", which will convert the plot back to a Scatter plot.

The plot type can also be changed by using the drop-down menu at the lower left corner of the dialog box. If the Line+Symbol type is selected then buttons for both **Symbols** and **Line** appear at the top of the dialog box, as shown below:
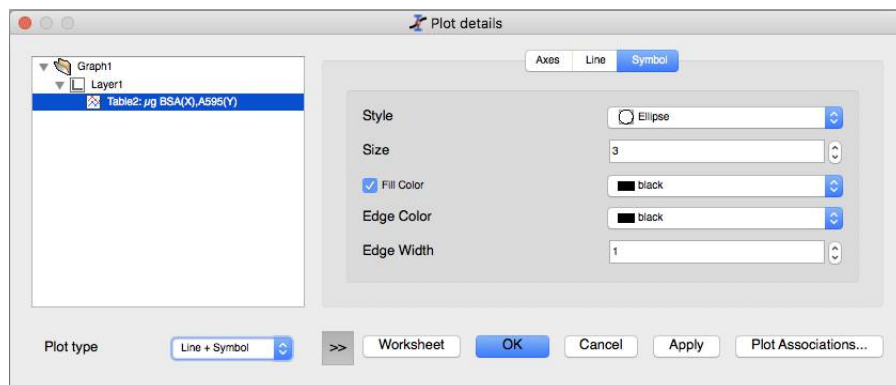
For our purposes here, the scatter plot is the best option, because we will be fitting curves representing mathematical functions to the data, and the line segments will be a distraction.

The default plot type can be set in the program preferences, which are accessed using the **Preferences** command in the **scidavis** menu. To set this preference, select "2D Plots" and click the **Curves** button. A variety of other default options can be found in the preferences. If you make this preference change, or others, on a computer of your own, they should be saved for whenever you use the program. On the Biology Department lab computers, the preference changes will be saved during your session, but they will be lost when you logout.

10

2. The axis scales can also be modified. In this case, I think that the plot would look better if the both the X- and Y-axes were to start with 0. To change the axis, select either **Scales...** or **Axes...** from the **Format** menu. Either command will open the **General Plot Options** dialog box, shown below:
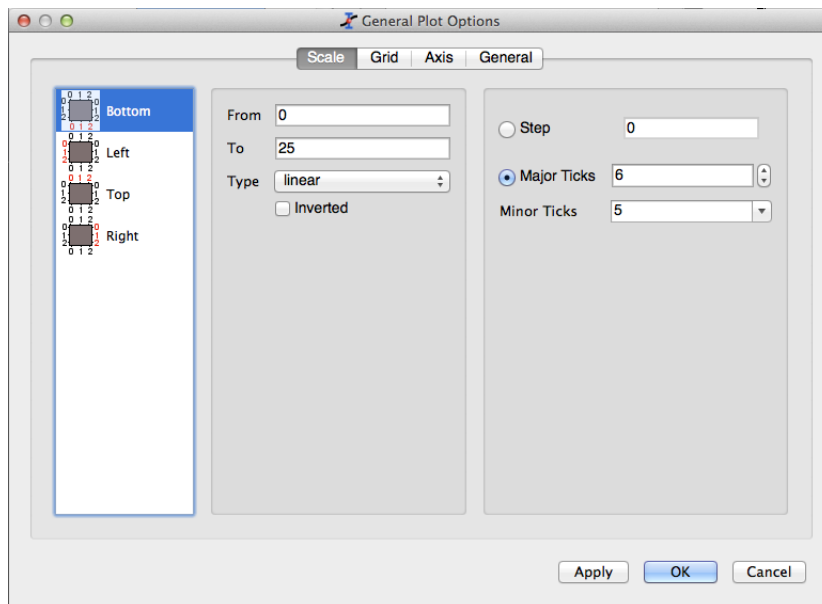

The white box to the left is used to select one of the four possible axes. By default, only the bottom and left axis are drawn in a new plot, but top or right axes can be added later. (An option in the preferences can be set to draw all four axes by default.)

First, we will work on the left axis, representing A_{595}. With the left axis selected, click on the **Scale** button at the top of the dialog box. This will bring up fields in which to enter the minimum and maximum values of the axis scale, as well as options for logarithmic or inverted scales, and options to change the number of major and minor ticks on the scale. The same options can be applied to any of the other axes that are displayed.

Horizontal and vertical grids can also be added to the graph, by selecting options that appear when the **Grid** button is clicked. Although they are sometimes useful, grids usually just add clutter to a graph and are discouraged.

3. The **Axis** button brings up a variety of additional options for each axis, including the axis name and the display of labels on the axis. This window is rather complicated and is shown below with four regions labeled for reference.:
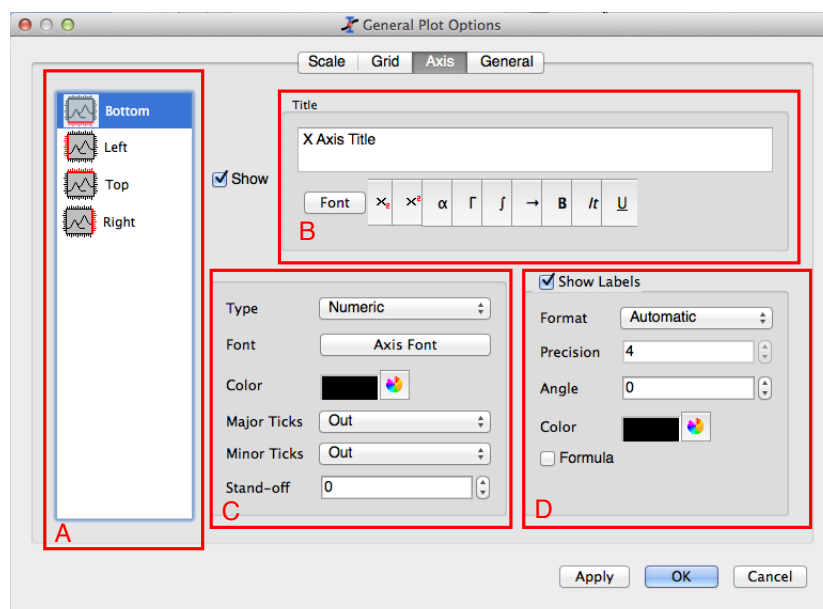
As described above, the white area in the box labeled A is used to select the axis to be modified.

The checkbox labeled **Show**, between the boxes labeled A and B, is used to turn on or off each of the axes. As discussed below, an axis can be turned on, but the labels hidden.

The region labeled B is used to modify the label on an axis. The text of the label can be edited in the text field, and the font style and size can be changed by clicking on the box labeled **Font**. The boxes to the right of the **Font** button can be used to format individual characters or to enter special characters:
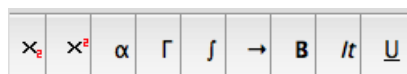
The two buttons on the left of this row are used to enter subscripts and superscripts, and the three on the right are used for bold characters, italics and underlines.

The four buttons in the middle, labeled α, Γ, $\int$ and $\rightarrow$, each open up a panel with special symbols that can be selected and entered into the label by clicking on them. Unfortunately, these special panels are often hidden behind the **General Plot Options** window, so that you may need to move the windows around a bit to find them.

You should use these options to label the X- and Y-axis "μg BSA" and "A$_{595}$", respectively.

The area labeled C in the illustration contains options for the number labels on axes, including the font and color, and the display of the ticks. (The actual placements of the major and minor tics are set using options in the **Scale** panel.)

The area labeled D has additional options for the labels on the axes, including whether or not the labels will be shown. Often, it is useful to show the top and right axes,

12

but hide the labels. The representation of numbers, in decimal or scientific notation can also be set, along with the number of decimal places to be shown.

There is also a checkbox labeled "Formula". Clicking this box opens a new text entry box that is used to enter a formula for calculating the values used to label the axis. For instance, typing the string "10*x" in the formula box for the bottom axis, will change the numbers to 10 times the values in the data window. This is a rather specialized option, but it could be useful for for adding a second axis, for instance the top axis, with values that are somehow related to the ones shown on the bottom.

4. The last dialog within the General Plot Options window is for changing the overall appearance of the graph, accessed with the **General** button and shown below:
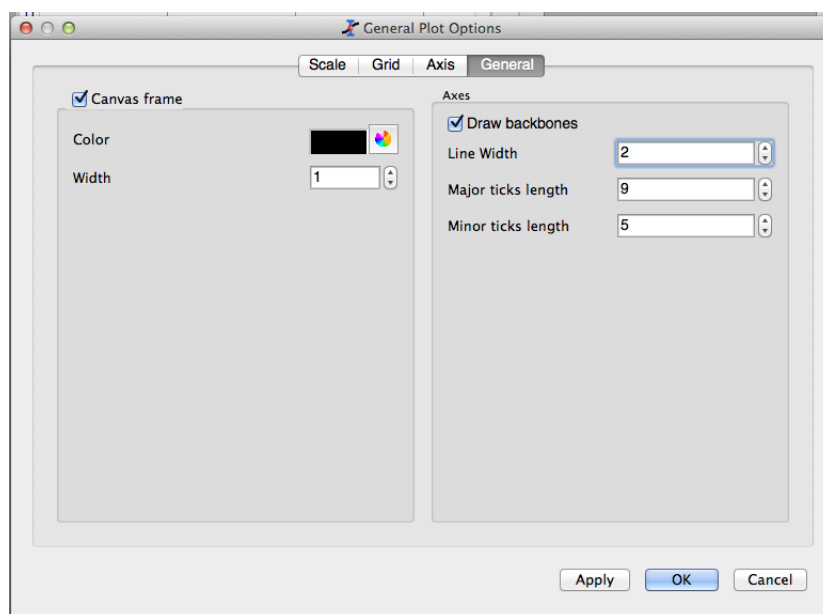
This box deals with the way in which the border lines of the graph are drawn. If a checkmark appears in the box labeled **Canvas frame** a box will be drawn outside of the plot and all of the labels associated with it. The color and widths of the lines used to draw this box can be changed using the controls below the checkbox.

On the other side of the dialog box, there are controls for changing the appearance of the axes and tick marks in the box, including the option to turn off the axes altogether. These options can also be set within the program preferences, so that they are used by default.

The best choices for the canvas frame and axes backbones may depend on how the plots are to be used and the file formats used to save the graphs. Some experimentation may be required! For most purposes, the best options seem to be to turn the axes backbones on with a width of 1 pixel.

5. The title of the plot can be changed by either executing the **Title...** command in the **Format** menu or by using the mouse to click on the title while holding down the

Control key and selecting **Properties** from the drop-down menu. Either option will open up a dialog box in which the title and is formatting can be edited.

6. By default, SciDAVis will create a legend for each curve in a plot. This can be edited or deleted by using the mouse to click on the legend while holding down the Control key and selecting either **Properties** or **Delete** from the drop-down menu. Choosing properties will open a dialog box for editing the legend.

After the changes suggested above, the plot should look something like:
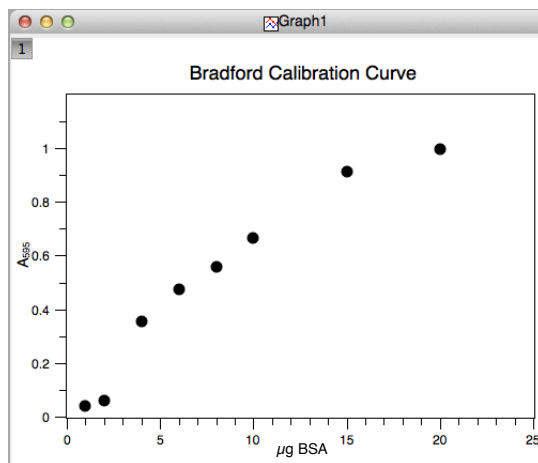
D.   Curve fitting

Fitting to a straight line

Often, we want to fit some mathematical function to our data. This might be just to provide a visual cue to make any trend in the data more obvious. Alternatively, we might believe that the data should be determined by a particular mathematical model and we want to test the predictions of the model or extract the values of parameters that help define the model (for instance, K_m and V_{max} in the Michaelis-Menten equation for enzyme kinetics.) In the case of the Bradford assay, we want a calibration curve that we can use to calculate protein concentrations from other measurements.

The simplest commonly used function to fit data is a straight line, which is defined by the equation:

$$Y = m \cdot X + b$$

where m is the slope and b is the Y-intercept (the Y-value corresponding to $X = 0$). When we say that we are "fitting" the data, we mean that we are finding values for m and b that, in some sense, give the best match between the experimental data and the values predicted by the equation. For a straight line, a perfectly reasonable way of doing this is to place a ruler across the graph and adjust its position until it seems to be as close to the data points as possible, intuitively giving all of the data points roughly equal weight. The slope and intercept can then be read from the graph. There are, however, at least two difficulties with this procedure. First, if the data display much scatter, then

drawing a line can become rather subjective, so that any two people are likely to draw slightly different lines, and any preconceived notions about the data might influence the result. Second, simply drawing a line doesn't provide any objective measure of how well the data and line actually match.

The most common technique used to fit data to a line or other function is the *method of least-squares*. This method is based on the idea that if the experimental data are determined by a linear function, then the values of each of the individual data points, X_i and Y_i, can be expressed as:

$$Y_i = m \cdot X_i + b + \epsilon_i$$

where m and b are the "true" values of the slope and intercept, and ϵ_i is an error term, presumed to be random, associated with each point. If we try to simulate the data using values we somehow choose for m and b , then the Y-value *predicted* for a given X will be:

$$Y_i' = m' \cdot X_i + b'$$

where Y_i' is the predicted value and m' and b' are the chosen parameter values. The difference between the observed and predicted Y values is called a *residual* and is given by:

$$\chi_i = Y_i - Y_i' = Y_i - (m' \cdot X_i + b')$$

In the method of least squares, each of the residuals is squared and then they are all summed to give a term called χ^2:

$$\chi^2 = \sum_{i=1}^{N} \chi_i^2$$

where N is the total number of data points. The actual value of χ^2 will depend on the values of the data points and the specific values m' and b' that are chosen. Qualitatively, it seems reasonable that there would be values of m' and b' for which χ^2 has a minimum value for a given data set, as sketched below:
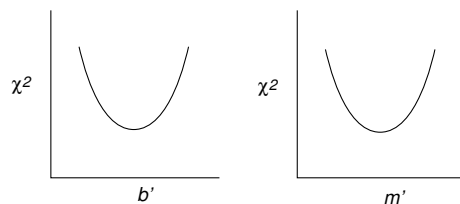

If either of the parameters is made either very large or very small, there will be huge deviations between the observed and predicted Y values.

For a straight line, it is relatively easy to derive equations that define the values of m' and b' that minimize χ^2, and these equations are pre-programmed into many handheld scientific calculators and computer programs. The process of fitting a line to data by least

squares is often referred to as *linear regression*, and the resulting equation is sometimes called a *least-squares line*.

In SciDAVis, a line can be fit to a data set by selecting the item **Quick Fit** in the **Analysis** menu and then choosing **Fit Linear** from the drop-down menu, as shown below:
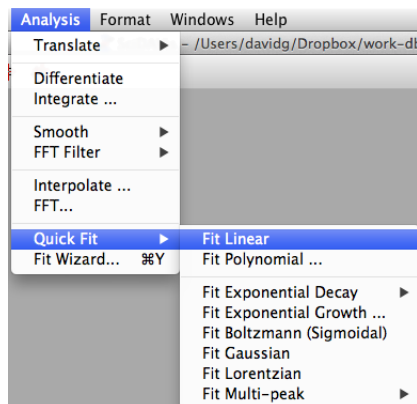
If there is only one data curve on the plot, the program will go ahead and fit the data and add the calculated line to the graph. If there are more than one data curves, a dialog box will appear, allowing you to choose the data set to be fit. In either case, the fit line should appear on the graph:
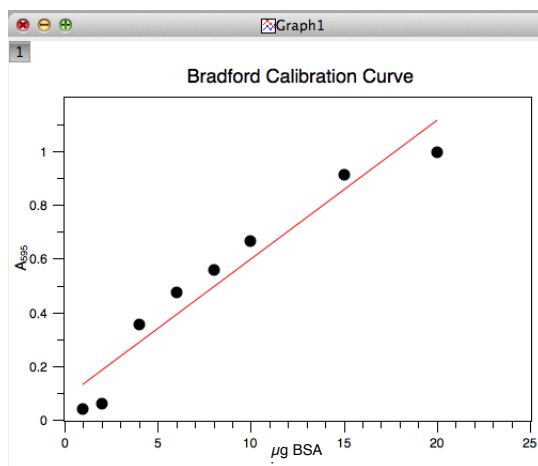

The properties of the line displayed in the graph can be modified by opening the **Plot Details** dialog box, which is accessed with the **Plot...** item in the **Format** menu. When this box is now opened, there will be a new curve identified in the box on the right hand side, as shown below:
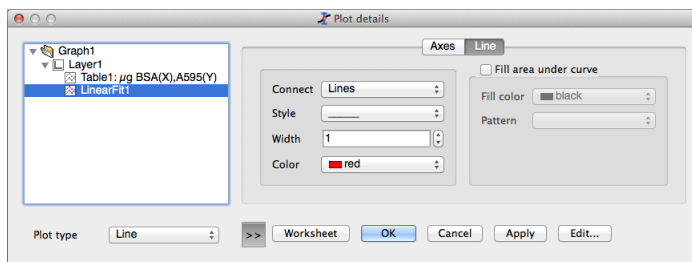
When the new curve is selected, and the **Line** button is pressed, as shown above, the style (dash pattern), width and color of the line can be changed.

The fit curve, or other curves on the plot, can be removed using the **Add/Remove Curve...** command in the **Graph** menu, which opens a new dialog box:
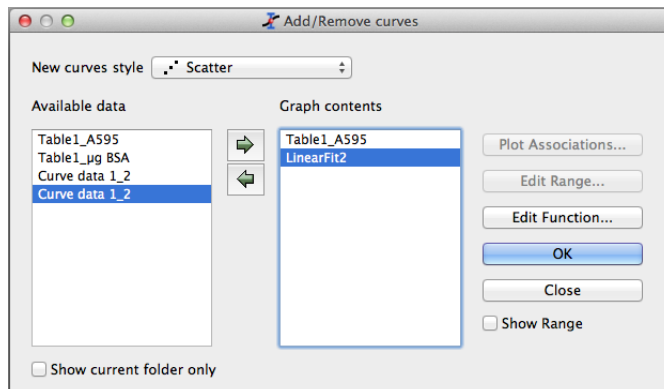

This box has two panes. The one on the left lists data columns that are available for plotting, whereas the one on the right lists the curves already on the plot. To remove a curve, select an item from the righthand panel and click on the button with an arrow pointing to the left. Conversely, to add a curve, select an item on the left and click on the button pointing to the right. After making the changes, click on the **OK** button.

In addition to drawing the fit line on the graph, a new window should appear, labeled **Results Log** and containing the results of the fit, including, the formula of the fit function, the estimates of the parameters (with error estimates, discussed below), and the value of χ^2:
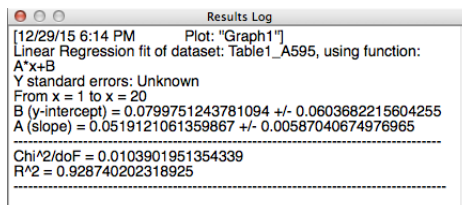
An additional parameter, R^2, is also shown in the log. This parameter, called the *coefficient of determination*, is a measure of how well the data are fit by the line. If calculated correctly, the value of R^2 should always lie between 0 and 1, with larger values representing better fits. Quantitatively, R^2 represents the fraction of the total variation in the Y values that are predicted by the fit line. Thus, if $R^2 = 0.9$, then the equation accounts for 90% of the variation, and the remaining 10% must be accounted for by other factors, including experimental inaccuracies. It is important to realize, however, that demonstrating a good fit to a particular function does not prove that the physical process is determined by a particular model. On the other hand, the failure to demonstrate a good fit can sometimes help rule out possible models.

The SciDAVis preferences can also be set so that the plot parameters are automatically pasted onto the plot. This option can be set by selecting **Preferences** from the **scidavis** menu to open the preferences dialog box:
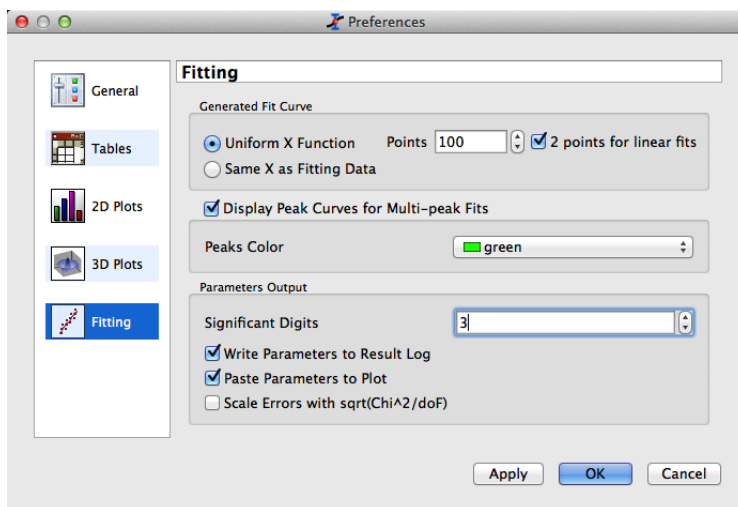
Preferences related to different aspects of the program are set by selecting from the boxes on the right. To automatically add the fit parameters to plots, click on the indicated check box. It is also a good idea to change the number of significant figures used for the parameters, from the default 15 to a more reasonable 3.

Fitting to polynomials

The same mathematical method can be used to fit a variety of other functions, such as polynomials of the form:

$$Y = m_0 + m_1 \cdot X + m_2 \cdot X^2 + \cdots + m_N \cdot X^N$$

A polynomial in which the largest power of X is N is referred to as an Nth-order polynomial. Very likely, your Bradford calibration curve displays some downward curvature, and you may be able to obtain a better fit using a polynomial.

To fit a polynomial to your data, again select **Quick Fit** from the **Analysis** menu and choose **Fit Polynomial...** from the submenu, to bring up the dialog box shown below:
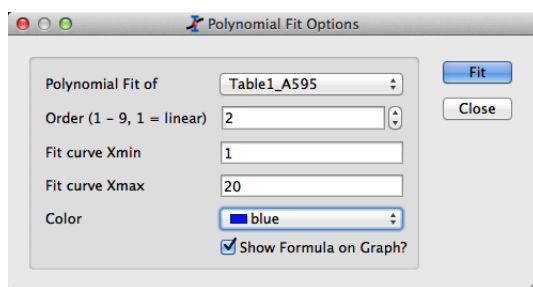

This box allows you to choose the data set to fit and the order of the polynomial to be used. It also allows you to select the range of data to be used in the fit, and the color of the curve. First try using a second-order polynomial (a quadratic equation). Since we already have a red curve for the straight-line fit, we will use blue for this curve.

18
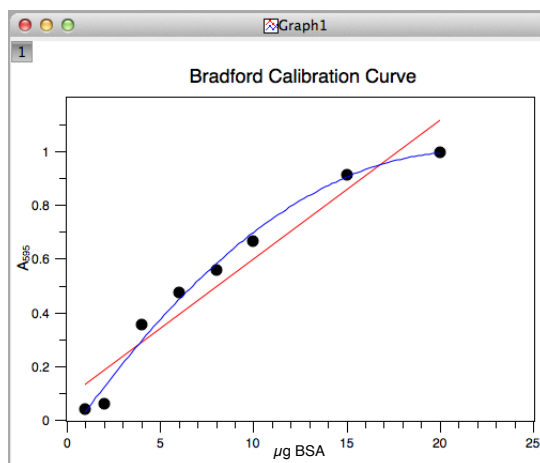
You will likely find that this gives a slightly better fit, as indicated by the parameters in the log:
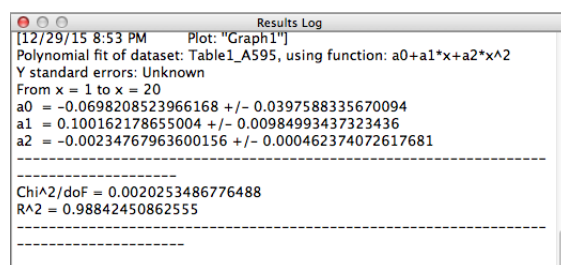
Note that χ^2 has decreased, and R^2 has increased, compared to the values for the linear fit. In general, using a more complex equation with more parameters will almost always give a better fit to a given data set. In fact, with N data points, one can usually obtain a *perfect* fit ($R^2 = 1$) using a polynomial of order $N - 1$. The result, however, is usually pretty meaningless, as illustrated below for a Bradford calibration curve with 8 points:
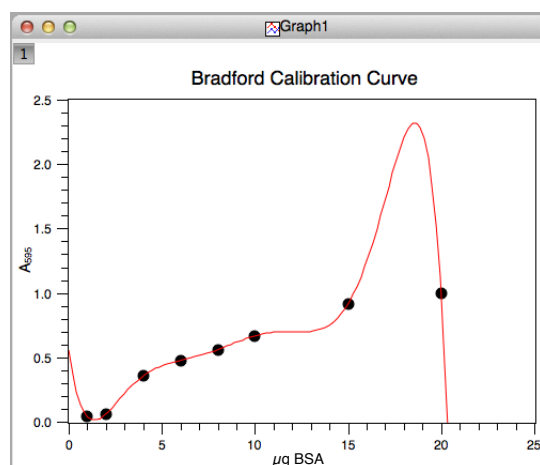
This extreme example illustrates one of the fundamental issues that arises in curve fitting. One can almost always obtain a better fit using more parameters, but it is not always clear when using a more complex function is justified. In the case of the Bradford data

used here, there appears to be a systematic curvature, and so using a non-linear function is probably justified. But, you don't want to end up with an elaborate function that is determined by the relatively small errors in each data point. In general, the number of parameters that are extracted should be several times smaller than the number of data points.

Non-linear least-squares fitting

The **Quick Fit** submenu of the **Analysis** menu also includes built-in curve-fit routines for other simple functions, including exponential functions. More importantly, however, the program includes a tool for fitting data to almost any mathematical function. As an example, we might want to fit the Bradford data to a function of the form:

$$Y = \frac{a \cdot X}{b + X}$$

This function, with parameters a and b, defines a rectangular hyperbola, for which Y increases almost linearly for small values of X and then approaches an asymptote when X is much greater than b. (If this equation isn't familiar to you, it will be soon!)

The hyperbolic function differs in an important way from the linear and polynomial functions in that Y is not a linear function of the parameter b (though it is a linear function of a). Because of this, the relatively simple method used to fit a line or polynomial cannot be used for a hyperbola, or for most other functions. Instead, an iterative procedure must be used. In this procedure, the program is given initial estimates of the parameters and then these are modified to minimize the sum of the squares of the residuals (χ^2). The procedure is repeated until χ^2 does not decrease further. The most commonly used method for this is called the Levenberg-Marquardt algorithm, which is implemented in SciDAVis, Kaleidagraph and other programs. Because it can be applied to functions that are not linear with respect to the parameters, it is often called *non-linear least-squares* fitting, which is a bit confusing because linear least-squares fitting can be applied to functions more complex than a straight line provided that the function is linear with respect to its parameters, as in the case of a polynomial.

The following steps describe how to do a non-linear least-squares fit with SciDAVis.

1. At this point, it may be useful to start with a fresh graph of your data. But, if you have spent time modifying the appearance of the graph, for instance by resetting the scales of the axes and their labels, you probably don't want to do all of that over again. Instead, you can create a new copy of an existing graph, by selecting the **Duplicate** item in the **Windows** menu. Then, you can remove the existing fits from the new graph, using the **Add/Remove Curve...** command in the **Graph** menu.

2. Select **Fit Wizard...** from the **Analysis** menu. This will open a dialog box that looks like:
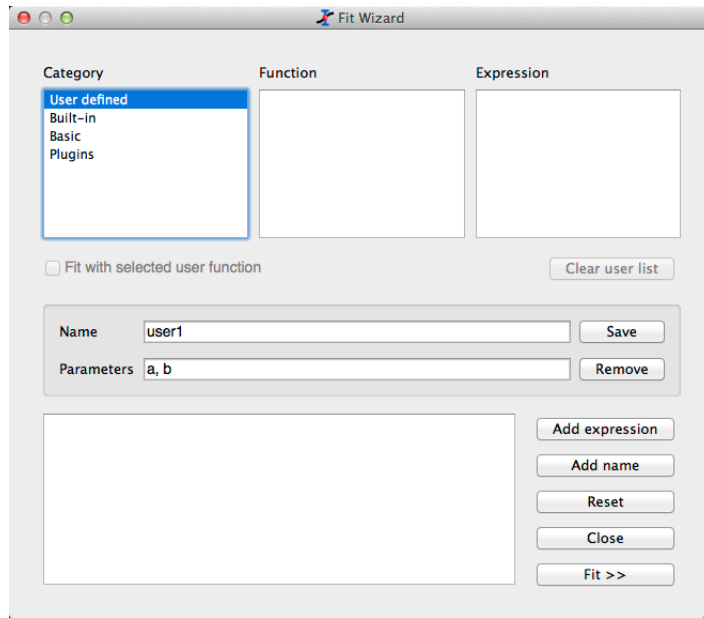
The top part of the box provides a list of built in functions as well as a list of basic functions that can be used as elements in user-defined functions. If previously defined user functions have been saved, these can also be selected.

When a new fitting function is created, the program initially defines two parameters, "a" and "b", and displays them in the **Parameters** box. But, other numbers of parameters can be used, and other names can be assigned to them. The function can also be given a name. The expression defining the function is then typed into the larger box at the bottom. The symbol "x" is always used as the independent variable in the expression. If the **Save** button is clicked, the name and expression are added to the collection of user-defined functions at the top of the window, and can be reused later. The window should now look like:
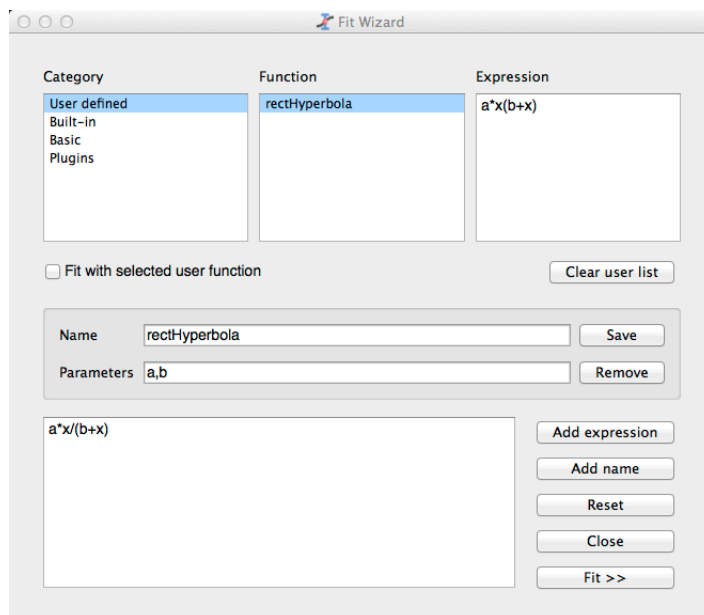
3. Next, click on the button labeled **Fit >>**, which will open another panel of the **Fit Wizard** window:

This window offers a number of options, some of which will be used later in the course. At the top of the window, there is a drop down menu, from which the data to be fit are chosen, if there are more than one set on the plot.

The most important parameters are the initial guesses for the fit parameters. For some functions, including the rectangular hyperbola, the algorithm is not very sensitive to the initial parameter values. In other cases, however, the initial guess may have to be fairly close to the final value, and an initial attempt to fit the data may fail. This will usually result in a message in the **Results Log** box saying something like "Status: Failed", or you may get a fit that is obviously wrong, such as a straight horizontal line. If this happens, first check to be sure that you have defined the function properly, and then try using initial parameter estimates that you think may give better fits to the data. Usually, you should be able to do some sort of "eyeball" fitting that will give you a reasonably good starting point. For the rectangular hyperbola, the default initial values of 1 for a and b will usually be fine.

4. Before doing the fit, there is one other option that should be set by first clicking on the button labeled **Custom Output>>**, which will open another pane:
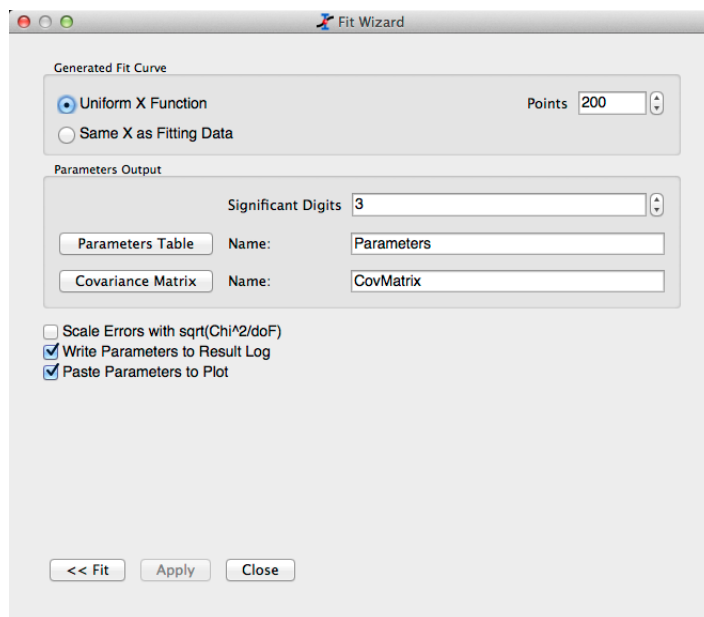
If they aren't checked already, click on the checkboxes labeled **Paste Parameters to Plot** and **Write Parameters to Result Log**, as shown above. It is also a good idea to change the number of significant digits to 3, if this hasn't already been done in the program preferences. Then, click on the <<**Fit** button, to return to the previous pane.

5. Finally, click on the **Fit** button in that pane. If successful, the fit curve should appear in the plot window, and the results will appear in the log window. In addition, the values in the boxes for the initial guesses will change to the fit values of the parameters. When the **Fit Wizard** window is closed, the parameters will also be pasted to the plot window (if that option was chosen, as described above).
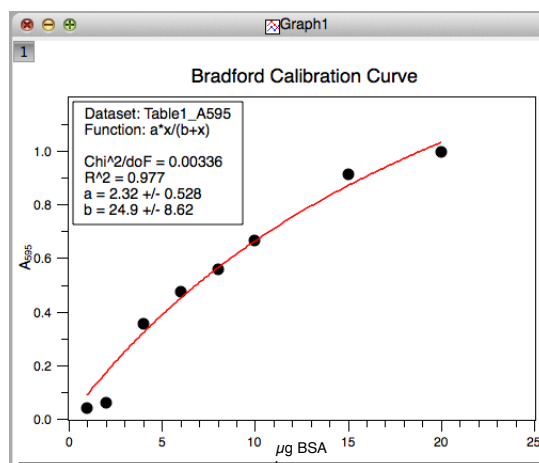

By default, the fit curve will be drawn over the range of X-values defined by the experimental values. Sometimes, though, it is nice to extend the curve over a wider range. The range can be modified by first selecting **Plot…** from the **Format** menu

to bring up the **Plot Details** dialog box. Then select the plot corresponding to the non-linear fit from the box on the right and click on the **Line** button, as shown below:
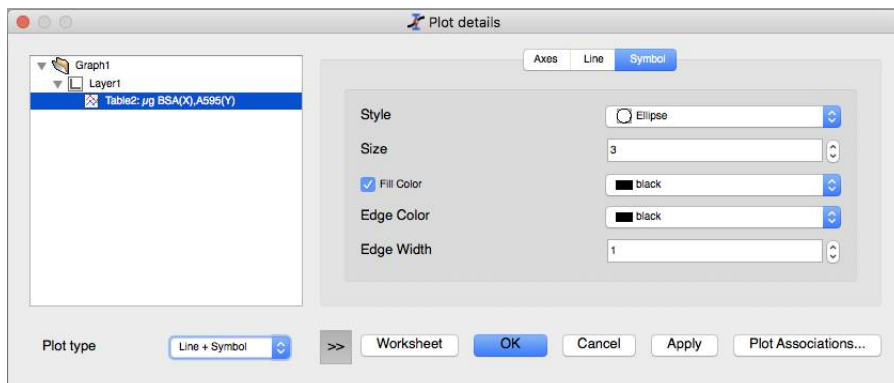
Next, click on the button labeled **Edit...** button, to bring up the **Edit function** box shown below:
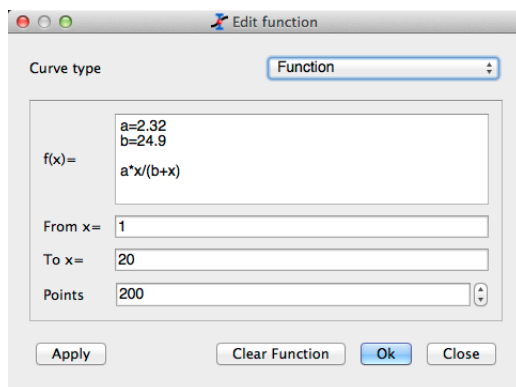
SciDAVIS can be used to plot mathematically defined functions, as well as data derived from other sources, and the display of fit curves is just one case of this general functionality. The **Edit Function** window can be used to change the parameters or even the definition of a previously plotted function, but that is not what we want to do here. Instead, we will just change the range of X-values over which the function is calculated and plotted, in this case from 0 to 25. After clicking **Apply** and **OK**, the range of the curve will be modified to the new values and the dialog box will close.

With this and other manipulations, you may find that the range of the axes in the plot will automatically change. This behavior can be rather annoying, but it can be turned off in the preferences dialog, under the options for 2D Plots. Uncheck the box labeled **Autoscaling**.

The output for all of the curve fits in SciDAVis also lists estimated errors for the fit parameters. These estimates are roughly equivalent to an estimate of the standard error of the mean (SEM), such as you calculated for your pipette calibration experiment. In the case shown above, you will see that the errors are actually quite large. The major reason for this is that the parameter a reflects a maximum value of Y that is approached

24

when X becomes much larger than b, which doesn't actually happen in this data set. At lower X values, Y depends mostly on the ratio of a and b, and different combinations of the two parameters that maintain a similar ratio will yield fits that are almost as good. As a consequence, the uncertainties in a and b are rather large, even though the data display relatively little scatter, and the ratio of the parameters is actually fairly well defined.

A final advantage of the hyperbolic function is that it can be easily rearranged into a form that allows you to calculate X-values from Y-values:

$$X = \frac{b \cdot Y}{a - Y}$$

This is particularly useful for the Bradford assay calibration curve, since you want to determine the protein concentration of an unknown sample from the measured absorbance. You could do this with the quadratic function as well, but it is much less convenient.

The use of non-linear least-squares to fit data has many applications. We will use it extensively to analyze the data from enzyme kinetics experiments. In this case and others, the method allows one to fit data directly to an equation, rather than using some rearranged form of the equation that is usually far less intuitive and can lead to undesirable artifacts, as we will see when we discuss methods for analyzing enzyme kinetics.

Deleting points from a fit, but still showing them on the graph

There is often a temptation to remove data points from a graph in order to "improve" the fit to a function. In general, this is something to be very wary of! But, there are times when it may be appropriate. For instance, there may be reasons to believe that some data values exceed the range where the function is valid, or a data point may lie so far from the trend defined by the other points that one suspects that something went wrong with that measurement. In these cases, it may be legitimate to exclude points from the fit, but honest reporting requires that those points also be shown on the graph and be clearly identified. To do this within SciDAVis, it is necessary to split the data into separate columns, representing the data that will be used for fitting and those that will be excluded, but still shown on the graph.

As an example, the data shown above for the Bradford calibration data suggests that there is a systematic downward curvature in the data as the protein concentration is increased. It may, therefore, be reasonable to fit a straight line to only the data points for lower protein concentrations, *provided that* the fit line will only be used to analyze results from other measurements in this range.

We begin by creating two new columns (using the **Add Column** command in the **Table** menu) and copying the absorbance values to each of the new columns:

The two new columns have been named "A595-fit" and "A595-excluded."

Next, delete the data to be excluded from the column to be used for fitting, and delete all but these values from the column of excluded data. This can be done either by deleting individual values or by selecting a range of entries and using the **Delete Selection** command in the **Edit** menu. The data window should now look like this:

Now, generate a new graph, with both the data used for the fit and excluded data points shown. Although you may be tempted to add the new points to an existing graph, this can lead to some perplexing behavior and may even cause the program to crash.

Finally, the plot symbols can be changed in the **Plot details** window, which is opened by selecting **Plot...** in the **Format** window. (See page 9.) A good choice here is to use filled symbols for the data points that are to be used for fitting and empty symbols, of the same size and shape, for the excluded points. The graph then looks like:
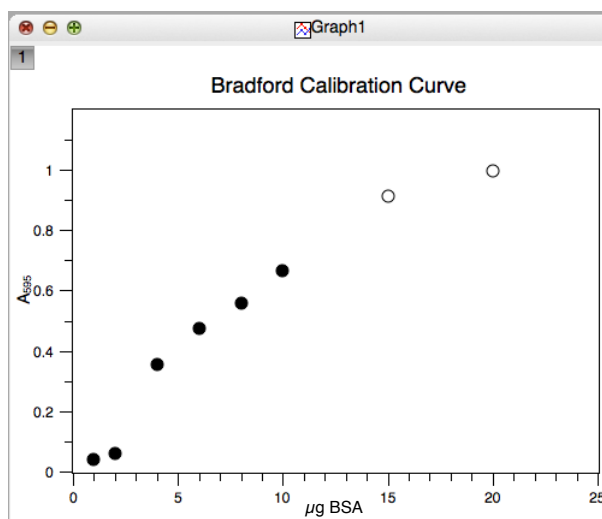
Now, the linear curve fit can be applied to the data without the excluded points:
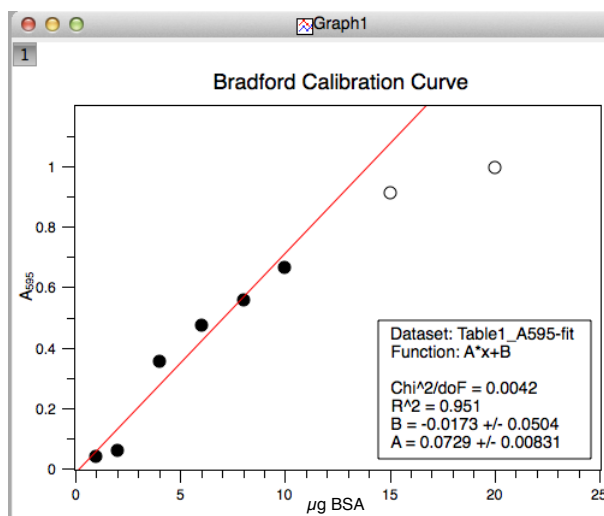

Note that the R^2 value is a bit larger than when all of the data were used, but the fit line now diverges quite far from the last two points. The important point, however, is that the scientist must be very clear in identifying the points that were used for the fit, as well as those that were excluded.

E. Exporting graphs

After creating and modifying a graph in SciDAVis, you will usually need to export it as a graphics file to use it for other purposes, including in your lab reports for this class. The lab reports should include both the SciDAVis project file and images of the graphs, in the png (Portable Network Graphics) format described below. For other purposes, you may want to use other image file formats.

The steps for exporting a graphics file for a graph are described below:

1. First, select the graph window that you want to export.

2. From the **File** menu, select **Export Graph** and then **Current...** from the submenu:

27
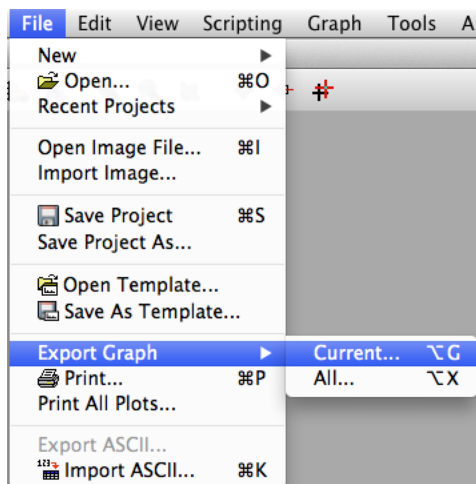
3. You should then see a dialog box like the one shown below:
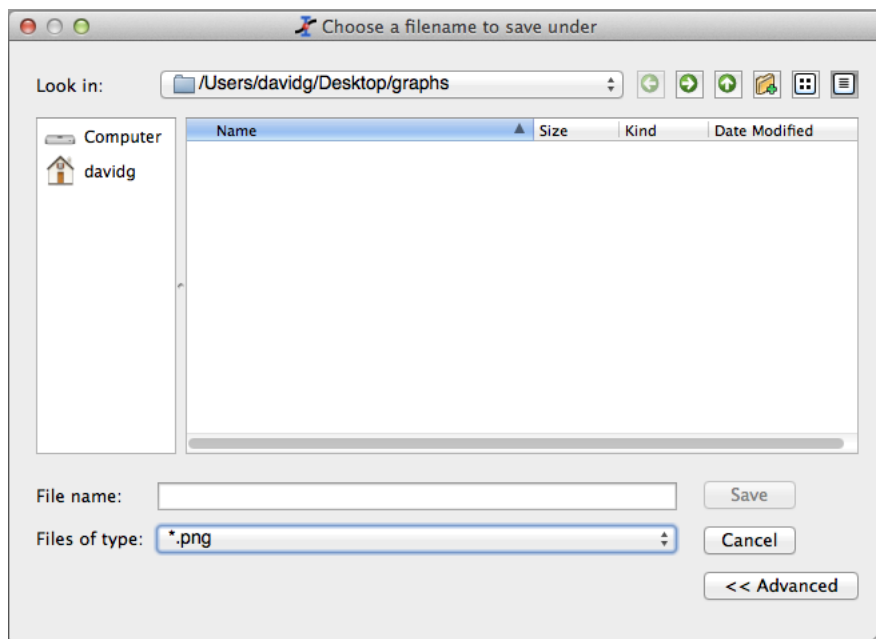
Although this file save dialog box may be slightly different from the ones usually seen on the Mac (or other operating systems) it has most of the same basic functionality. You can navigate to different folders, create new folders and specify the name for the new file.

At the bottom of the window there is a drop-down menu used to select the file type to be used, which are identified by standard file name extensions. Without describing all of the options, some of the most commonly used formats are:

- .png - Portable Network Graphics. This file format is representative of the general class of *bit-mapped* image formats. The important feature of this kind of format is that the image is represented as individual pixels, such as generated by a digital camera. Other formats of this type include the ubiquitous jpeg and tiff formats. Unlike jpeg files, png files are saved without any loss of image quality, but they are compressed to minimize storage space. The png format is also completely open, and it is becoming a preferred format for sharing images when it is important to maintain the original data in the image, some of which is lost each time a jpeg file is saved as a new file. png files can be viewed directly in LabArchives, which is why they should be used here. A general limitation of bit-mapped images is that their resolution is limited to that of the original image. Although there are methods for increasing the resolution of an image, that is the number of pixels used to represent it, the result is rarely as good as an original file saved with the same number of pixels.

- .eps - Encapsulated PostScript. In contrast to png, jpeg and tiff files, eps files are an example of *vector-graphics* formats. Files of this type contain mathematical descriptions of the objects that make up the image, including lines, squares and

28

text characters, as well as other shapes. Vector files can also include bit mapped images that represent part or all of the image. For drawings and illustrations (as opposed to photographs), vector-graphics files require less data to specify an image than a bit-mapped representation of the same drawing. More importantly, the mathematical representation of the image can be used to draw the image with whatever resolution the output device (usually a screen or printer) is capable of. For high quality graphs, such as required for publication, the use of vector graphics files is almost always preferred, and eps is probably the most widely used format of this type.

Another advantage of vector graphics is that the elements in the image can be individually edited with suitable programs. Adobe Illustrator is one program that can open edit many vector-graphics file formats. Another is Inkscape, which is a free open source program: https://inkscape.org. Although the user interface for Inkscape is not quite as polished as those of some commercial products, it is a very capable program.

The eps, ai (Adobe Illustrator) and pdf formats were all developed by Adobe and are all based on the PostScript computer language, which was originally developed for specifying the output of high-resolution printers. The eps format, in addition to specifying the graphics, can also specify the box enclosing the graphics, and it is usually the best format for transferring vector-based images from one program to another, for instance inserting images in a word-processor file.

Unfortunately, as of version 2.8, SciDAVis does not support direct export of eps files, even though the option is listed in the Save dialog box. If this option is chosen from the drop-down menu, the image will be saved in the closely related pdf format described below. The resulting file can be opened a Illustrator or Inkscape, edited and then saved as an eps file.

- .pdf - Portable Document Format. This is the familiar format for sharing documents with minimal dependence on specific software. These files can be opened and viewed in the free Adobe Reader program, the Apple Preview program and a variety of other programs, both free and commercial. As noted above, the pdf files exported from SciDAVis can be opened and edited in Illustrator or Inkscape. However, many of the individual objects making up the image, including type characters, are made up of small line segments, making it somewhat difficult to edit those objects. Sometimes it is easiest to remove axis labels and other text and then add them back with the drawing program.

- .svg - Scaled Vector Graphics. This is another vector-graphics format and was developed by the World Wide Web Consortium and has the virtue of being supported by an open community. It is also the default file format for the open-source program, Inkscape.

  As of version 2.8, the svg files exported by SciDAVis suffer from a variety of artifacts, and are difficult to edit in Inkscape or Illustrator. Thus, the pdf export option appears to be the most practical way to generate vector-graphics files from SciDAVis.

Use the drop-down menu to select the format, and then type in a name for the file. The program will automatically add the proper suffix.

4. For some of the file formats, the button labeled <<**Advanced** is activated, and when it is clicked additional options appear in the window. For some file formats a setting for the image quality or resolution appears.

5. After the file format, name and location of the file have been specified, click on the **Save** button to create the file.

6. As mentioned on page 11, the settings for specifying the canvas frame and axes backbones, in the General Plot Options dialog, can sometimes have unexpected effects on the appearance of the these features in exported image files. If the frame is not what you expect or want, it may be necessary to experiment with these settings.

F.   Other techniques

Although the sections above should cover the methods that you will need for analyzing the data from Experiment 2, there are a variety of other features of SciDAVis that may come handy later in the course, or in other contexts. Some of these are described below.

Automatically creating a series of numbers in a data window

Sometimes it is useful to create a series of numerical values for calculations in SciDAVis. For instance, Problem 2 of Experiment 5 requires generating a series of numbers representing pH values from 1 to 11 at intervals of 0.05. Typing such a list would be very tedious! But, SciDAVis provides tools to make it relatively easy, as described below:

1. Create an empty table, or use an existing one.

2. When a table is selected, on the left border of the project window, there should be some small icons:
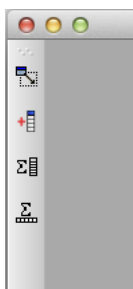
If these icons are not present, select **Table** from the **Toolbars** submenu in the **View** menu:
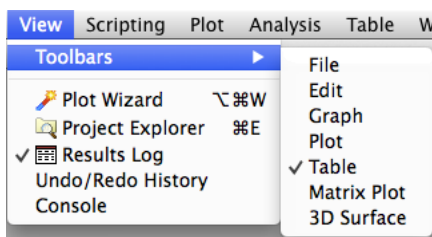
Clicking on the icon at the top should open a dialog box in which you can change the number of rows and columns in the table. For this case, set the number of rows to 201.

3. Select the first column in the table. In the **Table** menu, select the command **Fill Selection ▶ With Row Numbers**.
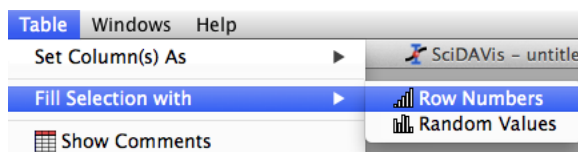
This should fill the selected column with the numbers 1 to 201.

4. Select the second column and open the formula panel. The formula below should fill this column with values from 1 to 11, with intervals of 0.05.
1+0.05*(col("1")-1)

5. You can then use the formula panel to calculate the contents of other columns from these values.

Importing an ASCII text file

Text files generated by other programs, including the one that you will be using to collect kinetic data automatically from the spectrophotometers in the lab (MacSpec), can be imported into SciDAVis for plotting and analysis. Before using this feature, however, it is necessary to know something about the structure of data files to be imported. As an example, if a MacSpec kinetics date file is opened in a text editor, it will look something like this:

```
#kineticsData
#MacSpec data document created: Sun, Mar 2, 2014
#
#Assays of freshly prepared affinity purified trypsin
#Trypsin diluted 500−x from nominally 0.5 mg/mL stock.
(min) 0 trypsin 10 μL  20 μL 40 μL 60 μL
0  0.000  0.000  0.000  0.000  0.000
1  0.000  0.008  0.014  0.018  0.044
2  0.000  0.017  0.029  0.036  0.057
3  0.000  0.026  0.043  0.054  0.104
```

4 0.000 0.034 0.058 0.072 0.152
5 0.001 0.043 0.073 0.090 0.199

The first five lines of the file contain information about the file and data, including any comments that the user saved with the data. The hash marks ("#") at the beginnings of these lines are used to indicate that these lines are to be ignored. This a common, but not universal, convention, and some programs can automatically ignore lines identified by this or other labels. When importing text files into SciDavis, the number of lines to be ignored must be specified.

The next line contains names for each of the data columns, with the first column representing a time in minutes and the others representing absorbance measurements.

The following lines contain the actual data. As shown above, the columns appear to be separated by ordinary blank spaces. However, the data files generated by MacSpec and many other programs use special (and ordinarily invisible) tab characters to separate data columns. This allows a distinction between the separation between data values and ordinary spaces, such as used in the column names in the example above. Because of this distinction, the space between, for instance, "(min)" and "0" is treated differently than that between "0" and "trypsin". The former is a column separator (a tab), whereas the second is not recognized as a column separator. Files that are structured this way are commonly referred to as *tab-delimited*. Columns of data can also be separated by commas or spaces, and SciDAVis can import files structured in these ways.

The key parameters that are needed in advance to import text data into SciDAVis are the number of lines to skip before reading data, whether or not the file includes a line with the names of columns, and the character used to separate columns.

To import a text file, use the **Import ASCII..** command at the bottom of the **File** menu (*not* the **Open** command, which only opens SciDAVis project files.) The **Import ASCII..** command will open a dialog box:
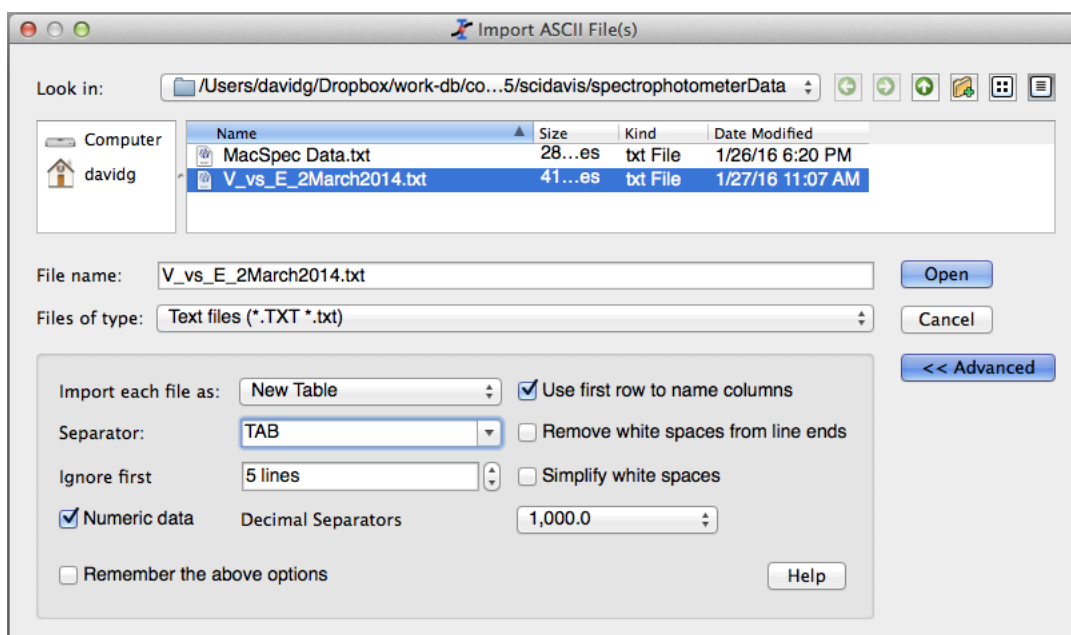
The top portion of the window is very similar to the one for exporting image files,

and is used to select the file for import. The drop-down menu labeled "Files of Type" allows the user to screen for file types, based on the file name extension (such as .txt, .dat or .csv), but the same basic text file type is assumed.

If the lower portion of the box shown above does not appear at first, click on the box labeled <<**Advanced**. This portion of the box includes a variety of options for specifying exactly how the text file is to be read. These include the separator to be recognized to identify multiple columns in the file (set to the tab character in the example above) and how to deal with the first lines of the file. In the example above, the option is set to treat the first line as names for the columns of data. There is also an option for skipping a set number of lines at the top of the file. In this case, it is set to 5.
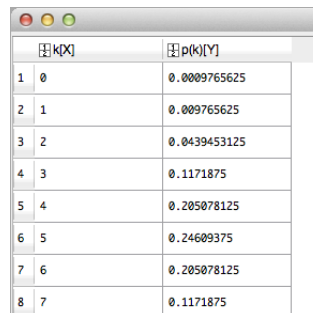
In the example above, the dialog is set to create a new data table for each file that is to be imported. Multiple files can be imported at the same time. There are also options to add the data to an existing table. After selecting one or more files in the dialog, the **Open** button becomes active, and clicking the button completes the action. The data should be entered into a new data table window within the project window.

The imported data can then be used to make plots and for other forms of analysis.
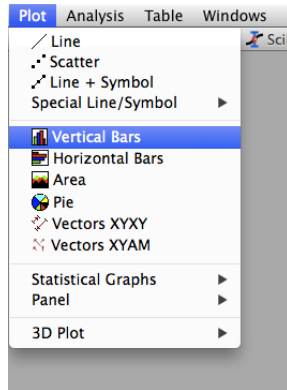
Bar graphs

SciDavis can also create bar graphs, but this option is not available through the commands and dialog boxes for adding, removing and adjusting other features of a graph. Instead bar graphs, with either vertical (much preferred!) or horizontal bars, must be created from a data window and using commands in the **Plot** menu that are available when a data window is open, as described on page 7.

To make a bar graph, first enter the data for the x- and y-axes in the columns of a data window, as discussed earlier and illustrated below for an example based on the binomial probability distribution function:

|	k[X]	p(k)[Y]
1	0	0.0009765625
2	1	0.009765625
3	2	0.0439453125
4	3	0.1171875
5	4	0.205078125
6	5	0.24609375
7	6	0.205078125
8	7	0.1171875

Make sure that the columns for the x- and y-axes are properly identified as such and are placed with the x-data column to the left. Then, select all of the rows with data in both columns and execute the **Vertical Bars** command in the **Plot** menu:

A bar graph like the one below should appear in a new plot window:
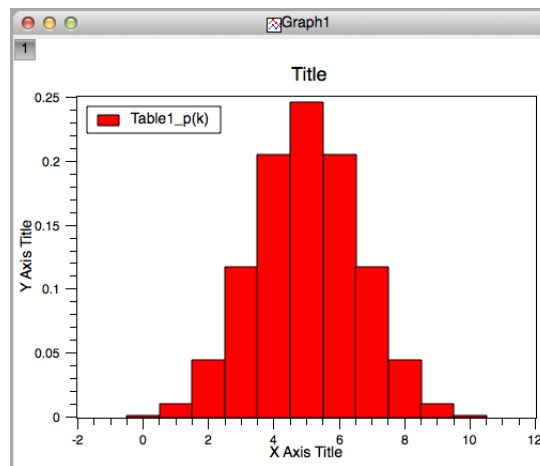
As with other plot types, the appearance of a bar graph can be modified by using the **Plot** command in the **Format** menu to open a Plot details window. For a bar graph, the window has a special pane for adjusting the spacing between the bars, as shown below:
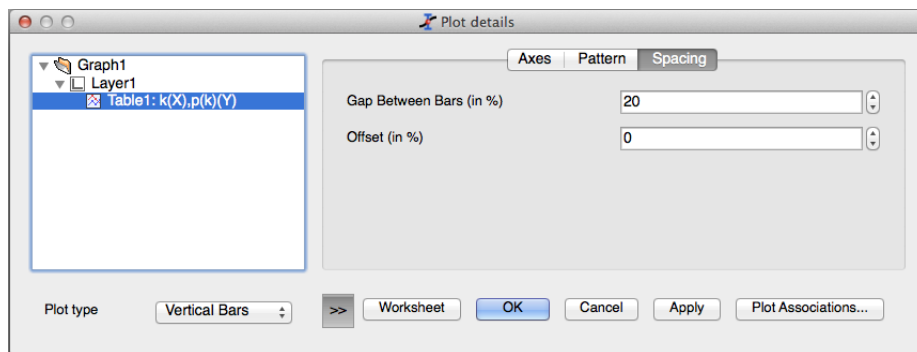

The space between bars can be adjusted to the improve the appearance of the plot, and sometimes it is useful to adjust the positions of the bars with the respect to the x-axis, depending on exactly how the values represented by the bars are related to the x-values. For instance, the bars may represent a value for a range of x-values, instead of a unique value. In these cases, it makes sense to adjust the positions of the bars so that they start with one x-value and extend to the next.

Text boxes on graphs

Depending on how the preferences for SciDAVis are set, the results of a curve fit may be automatically added to a graph. This information is added in the form of a text box, which can be modified. Text boxes can also be added manually to provide additional information.

To edit an existing text box, simply double click on it to open a dialog like the one shown below:
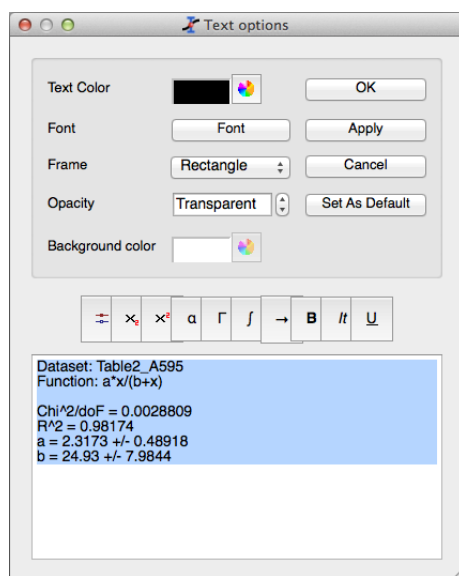
The text itself can be edited in the box at the bottom of the dialog box, and the appearance of the text and box can be modified in various ways using the controls in the upper portion of the dialog box.

One thing to be aware of is that the text entered in the box is not "soft wrapped" when it appears on the graph. That is, everything that appears as a single line in the dialog box, will appear as a single line in the box on the graph, expanding the width of the box as necessary. Thus, it is necessary to manually break the lines, using the return key, to manage the width of the text box in the graph.

To add a new text box to a graph, select the graph window by clicking on it, and then use the **Add Text** command in the **Graph** window. This command will open a dialog box asking if you want to add the text to an existing layer or create a new one. After making this choice a text cursor will appear, and moving this cursor to a position on the graph and clicking will open a Text Option box like the one shown above, in which you can enter your text and control its appearance.