ScanEdit User Manual

Table of Contents

Chapter 1. Functions by categories: What do you want to do? 1-1
1.0. Getting familiar with the program
1.1. Loading your data into the program,
1.2. Displaying spectra on the graphics screen. 1-2
1.3. Composing Figures
1.4. Plotting spectra to graphics files 1-3
1.5. Mathematical operations on spectra
1.6. Fitting spectra of mixtures
1.7. Smoothing spectra, taking derivatives, or integrating
1.8. Fourier transform
1.9. Convert "log epsilon" or "percent transmittance"
1.10. Scripting tasks for repetitive operations
1.11. Further assistance

Chapter 2. Getting Familiar with ScanEdit. 2-1
2.1. What is scanedit?
2.2 Two main windows-
2.3 Features of the main graphics window 2-2
2.4 Features of the console (trace selection window, or dashboard)

Chapter 3. Detailed Reference 3-1
3.1 The Graphics window
3.1.1 Graphics screen
    3.1.1.1 Mouse functions on the graphics screen
3.1.2 Buttons below the graphics screen
    3.1.2.1. "Drop Files here"
    3.1.2.2. Status bar- tells how many spectra are in memory 3-2
    3.1.2.3. A label displaying the Current data directory
    3.1.2.4. Exit button
    3.1.2.5. Console - Brings up trace selection window
    3.1.2.6. Plot All - Plot all spectra in memory on the graphics screen
    3.1.2.7 Clear Screen
    3.1.2.8. Zoom/Pan buttons

3.1.3 Dropdown menus on the graphic window 3-3
3.1.3.1 File Menu:
    3.1.3.1.1 Open (Load) (Kbd shortcut <Ctrl-O>)-
    3.1.3.1.2. Save- 3-4
    3.1.3.1.3. Save Ascii-
    3.1.3.1.4. Save All- Save everything in memory
    3.1.3.1.5. Import (Kbd shortcut <Ctrl-I>)
    3.1.3.1.6. Clear Memory. 3-5
    3.1.3.1.7. Print Comments-
    3.1.3.1.8 Redimension-
    3.1.3.1.8. Exit- standard file:exit, quits the program. 3-6

3.1.3.2 Display menu
3.1.3.2.1 Set Limits (kbd shortcut <ctrl-L>) 3-6
3.1.3.2.2. Clear screen 3-7
3.1.3.2.3. Plot all
3.1.3.2.4. Pot. . . .
3.1.3.2.5. Resize
3.1.3.3 "Operations" menu
3.1.3.3.1 Linear Combination
3.1.3.3.2 Print table: 3-9
3.1.3.3.3 Stack spectra
3.1.3.3.4 Differentiate 3-10
3.1.3.3.5 More derivatives-
3.1.3.3.6 Integrate
3.1.3.3.7 Smooth
3.1.3.3.8 Shift Spectra
3.1.3.3.9 Add Baseline Polynomials
3.1.3.3.10 Adjust spectral Range
3.1.3.3.11 Resample 3-11
3.1.3.3.12 Console - Brings up the trace selection window ("console")
3.1.3.4 SVD menu
3.1.3.4.1 SVD - Performs Singular Value Decomposition
3.1.3.4.2 Test EVD - test the Eigenvalue Decomposition 3-12
3.1.3.4.3 Reconstruct Orig. Spectra with N Components
3.1.3.4.4 Real spectra from fit to Singular timecourses
3.1.3.5 Fourier menu
3.1.3.5.1 Fourier Transform
3.1.3.5.2 Inv FT - Calculates the inverse
3.1.3.5.3 Damp a transform
3.1.3.5.4 Damped inverse FT
3.1.3.5.5 Frequency Spectrum
3.1.3.5.6 Draw Fourier (Frequency) axis
3.1.3.5.7 Inv. F.T. with resampling
3.1.3.5.8 Fourier components-
3.1.3.6 LSQ fit menu - 3-12
3.1.3.7 Disk-based 3-15
3.1.3.7.1 Make PS Figure
3.1.3.7.2 Subtract mean or best-fit line
3.1.3.7.3 LSinverse
3.1.3.7.4 Spectra × matrix 3-16
3.1.3.7.5 Run Script
3.1.3.7.6 Average
3.1.3.7.7 - other items are in development
3.1.3.8 Preferences - See Chapter 5, Customizing
3.1.3.8.1 change linemode - select drawing mode
3.1.3.8.2 Locate GS program -
3.1.3.8.3 Set Library directory -
3.1.3.8.4 Edit screen colors
3.1.3.8.5 Edit PSPlot colors 3-17
3.1.3.9. About ScanEdit

3.2. Trace selection window- ("console", "dashboard") 3-18
3.2.1- Selection List
3.2.1.1.Select
3.2.1.2. Selecting spectra with the dashboard 3-19
3.2.2 Controls at the bottom of the Trace selection window
3.2.2.1 Selection shortcuts ,
3.2.2.2 Dismiss
3.2.2.3 Changing the order of spectra 3-20
3.2.2.4 Remove
3.2.2.5 Identify
3.2.2.6 Edit comment
3.2.2.7 Plot (on screen)
3.2.2.8 unplot. 3-21
3.2.2.9 Autoscale
3.2.2.10 Statistics
3.2.2.11 Average and Sum 3-22
3.2.2.12 Clear Screen
3.2.2.13 Subtract from all
3.2.2.14 Subtract from some
3.2.2.15 Save (binary, .mtx, or .csv).
3.2.2.16 Multiply/divide
3.2.2.17 Make Figure 3-22
3.2.2.18 Add a constant 3-24
3.2.3 Drop-down menus on the Console (dashboard) 3-25
3.2.3.1 Resize.
3.2.3.2 superimpose spectra:
3.2.3.3 Set up linear combination.
3.2.3.4 Fourier
3.2.3.5 select all, deselect all
3.2.3.6 Set display limits
3.2.3.7 Three functions for converting the Y axis data

Chapter 4. Data Formats and Import 4-1
4.1 Native formats
4.1.1. Binary files. The primary format for
4.1.2. Ascii Text files. These have extension
4.2 Importing Other Formats 4-1
4.2.1 Importing Shimadzu UV-PC .spc files.
4.2.2. CSV (comma-separated values) files. (3.1.3.1.5, "import
4.2.2.1. Preparing the data.
4.2.2.2. loading the data: 4-3
4.2.2.3. Other examples: 4-4
4.2.2.3.1. Get the spectrum of Holmium oxide
4.2.2.3.2. A third example- a spectrum from the photochemcad package 4-5
4.2.2.3.3 Tabular text data - 4-6
4.2.2.4. Notes
4.2.3. JCAMP spectra, and other spectra with variable sampling interval 4-7
4.2.3.1 Background
4.2.3.2 Example 1: propionamide from NIST 4-7
4.2.3.3 Example 2: formatting an arbitrary spectrum 4-9

Chapter 5. Customizing ScanEdit 5-1
5.1. Getting ScanEdit - Download the file
5.2. Put the program in its place:
5.3. Make a shortcut on your desktop:
5.4. Program associations:
5.5. Environment preference settings: 5-2
5.6. Setting linemode - (3.1.3.8.1) This affect
5.7. Setting up the ghostscript program (GS) 5-3
5.8. Complications introduced by dynamic dimensioning

Chapter 6. Composing Figures and on-screen display 6-1

Chapter 7. Least squares fitting to a 7-1
7.1 Background
7.1.1 Simultaneous equations/Matrix inversion: If absorbance is
7.1.2 Classical Least Squares- 7-2
7.2 The importance of including baseline function(s). 7-3
7.3 How wide a spectral range should I use?
7.4 Importance of examining the fit. 7-4
7.5 Setting up for LSQ analysis 7-5
7.6 Other options in the LSQ fitting
7.7 Output files from the LSQfit procedure 7-6
7.8 More insight into the least-squares fitting

Chapter 8 - using SVD 8-1
8.1 Background 8-1
8.2 Procedure: 8-4
8.2.1 SVD.
8.2.2 Test EVD (eigenvalue decomposition) 8-5
8.2.3 Reconstruct Orig. Spectra with N Components
8.2.4 Real Spectra from fit to Singular Vectors 8-6
8.2.5 Test Data 8-7

Chapter 9. The Fourier menu: 9-1
9.1. Fourier transform.
9.2. Inverse FT. 9-1
9.3. Damp a transform. 9-4
9.4. Damped inverse FT.
9.5- Frequency power spectrum. 9-5
9.6- Draw Fourier (frequency) axis.
9.7. Inv FT with resampling.
9.8 Fourier Components.
9.9 Details of the Fourier transform routines. 9-5

Chapter 10. Smoothing and derivatives 10-1

Chapter 11. Scripting, to automate tasks in 11-1

Appendix A - Installation A-1

1. Getting scanedit to run
2. Update to latest "nightly build" A-2
3. Making scanedit convenient A-3

Appendix B. - Description of the Format of scanedit native files B-1
1. Binary files- Single spectrum each, can associate in series
2. Ascii ".mtx" files - single file contains collection of related spectra B-2

Appendix C - Troubleshooting
1. Why is the wavelength scale wrong?
2. Scanedit window turned white and "not responding"
3. Scanedit "crashed"

Chapter 1. Functions by categories: What do you want to do?

Data formats, importing spectral data, loading and saving data
Manipulating spectra: adding, subtracting, multiplying, and forming linear combinations
Graphical operations:
    Tables (or spreadsheet) of absorbance at selected wavelengths
    Printing postscript and pdf figures,
    Composing spectra to make figures - Chapter 6
Customizing scanedit: see Chapter 5
    default data and spectral library directories
    Font for PostScript output
    locating GSwin.exe for making pdfs
    change drawing mode (lines between vs dots at points)
    Editing the color pallets for screen display or postscript output
Linear least squares: Chapter x
    analyzing multi-component mixtures,
    superimposing spectra
Other mathematical operations:
    Linear combinations of spectra
    Smoothing, Differentiation and integration
    SVD (Singular Value Decomposition) analysis
    Fourier analysis
Installing the program - If it doesn't "just run", see appendix or separate pdf on this subject
Extract data from spectra (e.g. absorbance of a set of spectra at a list of wavelengths
Reverse calibration- determine the component spectra given training set
Perform singular-valued decomposition on a set of spectra
Use Fourier transform to show frequencies, filter spectra
Make derivative spectra, or integrate spectra
Show statistics of a spectrum or a set of spectra
Smooth spectra by sliding-window polynomial fit

1.0. Getting familiar with the program and running the tutorials.
Read Chapter 2

1.1. Loading your data into the program, all with the same sampling density and wavelength range, and saving modified or newly generated spectra.

ScanEdit natively uses two formats, described in section 4. Files can be loaded with the "*open*" and "*import*" options under the File menu, by dragging and dropping on the *"drop files here" pane* (3.1.2.1) of the graphics window, by dragging a file to a *shortcut icon* (chapter 5) for the program or (if you have set up fiole associations as decribed in Chapter 5) by double-clicking a file associated to scanedit. The latter two methods start a new instance of scanedit, change to the directory where the file is, and load the file(s).)

Various formats for spectral files are discussed in Chapter 4. ScanEdit can directly import Shimadzu UVPC (old style) *.spc files, and some JCAMP and Galactic ".spc" files. For anything else the easiest way to import the data is probably via a .csv file, which most spec programs can save. You need to edit

the .csv to insert a 1-row header, as described in section 4.2.2. The open and import functions are accessible from the file menu on the main (graphics) window, or by shortcuts ^o or ^i when the graphics window is in focus.

Re-sampling- In order to do arithmetic operations on digital spectra, they have to be sampled at the same wavelength, and to display correctly in scanedit the sampling interval must be constant across the spectrum. "Resample" under the operations menu changes the sampling interval between two constant-interval sampling densities. "Inv. Fourier with different sampling" does the same in a more experimental way. If the sampling is not at constant intervals, "Import JCAMP" will make a constant-interval approximation, even if the file is not JCAMP (see 4.2.3.3)

Files can be saved in either of the two native formats (Chapter 4). To save all spectra in memory in a series of binary format files, choose "save all" under the file menu. To save one or more selected spectra in binary format, or as a single ascii .mtx or comma-separated values (.csv) file, select the spectra in the Trace List and select the appropriate button in the "save" column at bottom of the console window.

1.2. displaying spectra on the graphics screen.

When spectra are loaded, the wavelength range is automatically adjusted to accommodate them and each spectrum is plotted. The absorbance scale is not automatically adjusted on loading. Absorbance and wavelength scales can be set by several options:

a. "set limits" under the Display menu (shortcut <ctl-L>, see 3.1.3.2) allows setting limits explicitly, or automatically set absorbance so that all spectra in memory are onscale over the chosen wavelength range.

b. The "Autoscale" button on the console window (3.2.2.9) allows setting absorbance so that selected spectra are onscale over a given wavelength range (and optionally setting the display to show only that wavelength range).

c. Use the right mouse button to drag a rectangle on the current graphics screen while holding down <ctl>, and that rectangle will be expanded to fill the screen (section 3.1.1.1)

d. using the "Zoom & Pan" buttons below the screen to zoom in and out or translate along the absorbance and or wavelength scale (3.1.2.8).

The screen can be cleared with the "clear screen" button on the lower right of the graphics window, and all spectra in memory can be plotted on the screen with the "plot all" button below it. Selected spectra can be plotted from the console window, selecting the spectra and clicking the "plot" button.

Spectra can be arranged vertically using the "stack spectra" option (3.1.3.3.3) under Operations menu to automatically displace successive spectra to prevent overlap, or by manually adjusting the vertical offset of selected spectra using the "Add Constant" button on the console window (3.2.2.18).

(The "move up" and "move down" buttons on the dashboard do not change the absorbance values or their position on the screen- they change the order of spectra in memory and in the list.)

1.3. Composing Figures-

See chapter 6.

1.4. Plotting spectra to graphics files for printing or inclusion in documents.
Chapter 6. Selected spectra can be "plotted" to a postscript file for making high-quality images. These can then be converted to pdf or, using a PhotoShop-type program, to gifs or tiffs. The dialog is activated by clicking "Make Figure" on the console window, see description and "composing figures". See (3.2.2.17)

1.5. mathematical operations on spectra.

The main tool for this is "Linear Combination" under operations (3.1.3.3.1). With this you can add a constant to a spectrum, multiply by constant, and add or subtract arbitrary multiples of up to 10 spectra. A new spectrum is created with the result, the original spectr(a) are unchanged. As a shortcut, in the console window select the spectr(a) to be involved, then choose "set up linear combination" (3.2.3.3) from the menu on that window. The "linear combination" dialog will be brought up populated with the selected spectra so you only need enter the coefficients for each.

For summing or averaging selected spectra to generate a new spectrum, see (3.2.2.16), these can also be done with the "linear Combination" dialog (3.1.3.3.1).

Multiplying or dividing all selected spectra by a constant(3.2.2.16)

Subtracting one spectrum from some or all of the spectra (including itself), see the corresponding items on the console (3.2.2.13,14)

For critically comparing the shape of two spectra that may be on different scales and have different offset, the option "superpose spectra" (3.2.3.2, and chapter 6) scales and offsets one spectrum to give the best (least squares) fit to another.

For smoothing, taking derivatives, integrating, Singular Value Decomposition, or Fourier transform operations, see 7 and 8 below.

1.6. Fitting spectra of mixtures -

This is the "Classical Least Squares" method of analyzing the spectrum of a mixture when spectra of the pure components are known. Detailed instructions are in section (3.1.3.6), and further advice and background are in chapter 6.

1.7. smoothing spectra, taking derivatives, or integrating
Use the corresponding options under Operations menu. Instructions in section (3.2.3.2), and background considerations in chapter 9

1.8. Fourier transform
Forward and reverse transforms, with options for filtering by frequency and displaying a frequency spectrum. Under the Fourier Transform menu of the main window, briefly described in section (3.1.3.5) with instructions and background in chapter 8.

1.9. Convert "log epsilon" or "percent transmittance" to absorbance or *vice versa*. (3.2.3.7)

1.10. scripting tasks for repetitive operations and to keep a record of operations.
Still in development- Chapter 11.

1.11 - Further assistance

If you can't find the answer in this documentation, feel free to contact the author:

eaberry1 at pacbell.net or eaberry@users.sourceforge.net

Check the forums: https://sourceforge.net/p/scanedit/discussion/749696/

Project web page https://scanedit.sourceforge.net/

Chapter 2. Getting Familiar with ScanEdit. This is a brief list of "what's where", and pointers to more detailed instructions in case the GUI is not self-explanatory.

2.1. What is scanedit?

ScanEdit is an editor for spectral data that facilitates viewing spectra and performing mathematical operations on them. Spectra can be loaded from computer files into memory, displayed on the graphics screen, operated on in various ways, and saved back to files. Some other functions operate directly from disk files, saving the results to disk. In memory individual spectra are held in "traces", or channels identified by trace numbers and "comments".

Each spectrum is associated with a descriptive comment describing the spectrum, which can be added or edited and will be save with the spectrum if it is saved to a file.

All spectra in memory at one time must be sampled at the same wavelength points, as this is required for most operations involving more than one spectrum. In case you are working with spectra at different sampling intervals, there are procedures for "resampling" the spectra to put them on the same sampling interval.

The program keeps track of two "current directories" for loading and saving files when no path is given, or for the default initial point for browsing to locate spectra. One is the "data dir" which is usually the current directory, where the spectral data being analyzed reside. The other is the "speclib" a directory containing standard spectra for use in analyzing the data. For the most part the speclib contains .mtx files with standard spectra for different types of analysis.

The idea is that you create a new directory for each new dataset, and that becomes the data dir (for example by loading files from there).But you may want to analyze the data using several different sets of standard spectra, which would be stored in speclib. However there is nothing that restricts you from using another model.

Two tutorials are included in the standard download (tutorials.zip), which also includes the executable and instructions for setting it up (but appendix 1 here is more up-to-date). Following is a brief overview of what you will find, details are in chapter 3, Reference.

2.2 Two main windows-

In addition to the graphics screen, there is the "console" (also called "Dashboard" or "Trace Selection window"). At startup, only the graphics screen is shown, and for some purposes this is all you need.. To see a list of spectra in memory, and for operations that require selecting spectra, open the console by clicking the button labeled console on the graphics screen, or with the keyboard shortcut <ctrl-D>.

Admittedly "console" may not be the best name for the Trace selection Window. The notion was that a console is a panel with a lot of controls and indicators. In modern computer parlance console seems to mean command line window for typing commands and receiving notification from the computer system. So we may be switching entirely to "dashboard" in the future. Other suggestions are welcome.

2.3 Features of the main graphics window

a. Drop-down menus from the top border - Most of the functionality that does not require selecting spectra (and some that does) is implemented here.

b. Graphics screen- where the spectra are plotted.

Controls at the bottom of the screen:

c. Drop Files here- Native format files (binary, .mtx), (old)Shimadzu .spc files, and .csv files can be dragged here to load. See Chapter 4 "Data formats"
This field is also used with the right-mouse button to display wavelength, absorbance at cursor positioned on screen.

d. Status bar- tells how many spectra are loaded and how many spectra of the current size would fit in memory.

e. Current data directory - Where files will be loaded and saved if no path is given.

Exit button - to quit the program. Alternatively ctrl-Q or file:exit

Zoom/Pan buttons – Zoom in and out, move up and down the absorbance or wavelength scale. See reference.

Console - Brings up trace selection window (console, dashboard) for listing the spectra, selecting, and many operations on selected spectra.

Plot All - Plot all spectra in memory on the screen

Clear Screen - Erase spectra and redraw frame

Legacy menu buttons- 3×3 array of numbered buttons – you probably don't need them. Left over from initial attempt to recreate the menus of the old DOS scanedit program. Never fully implemented, almost all the functionality is available now through the drop-down menus, and these buttons will be gone in the next major version (2.0).

Click, drag, and drop functions of the graphical window- Display Absorbance and wavelength of a point, identify trace passing through a point, select an area to zoom in on.

2.4 Features of the console (trace selection window).

Trace List- Lists the traces currently in memory, one to a line. Visually this consists of 3 parts: a colored swatch with the trace number, a selection box, and text giving the "descriptive comment" associated with that trace. The color swatch is the color used to plot that trace on the graphics screen, helping to identify each trace on the screen. The selection box is used for marking spectra for operations.

Drop-down menus on the trace selection window- resize, superimpose spectra, set up linear combination, convert %T=>A, LogA=>A, A=>logA

Buttons along the bottom of Trace Selection Window- These are the main controls of the dashboard; mostly acting on selected spectra in memory, or to select spectra. Buttons to select spectra (all, none, down, inverse), reorder traces in memory (move up, move down, swap, insert), remove (delete), identify, edit comment, plot, unplot, autoscale, statistics, average, sum, clear (graphics) screen, subtract from all, subtract from some, save (binary, .mtx (ascii), csv), make figure, add constant.

Chapter 3. Detailed Reference

3.1 The Graphics window

3.1.1 Graphics screen- Most of the area of the graphics window is taken up by the graphics screen, where the traces are plotted. The area is outlined by a rectangular line with tick-marks on all four sides. A scale on the left indicates the absorbance range, and a scale on the bottom indicates the wavelength range. Methods for adjusting the scale were listed in 1.2.

Spectra are automatically plotted when they are loaded, but you can clear the screen and plot selected spectra with controls on the graphics window or the dashboard. Most functions that generate new spectra plot them automatically. By default all spectra are replotted when the scale is changed, but options on the screen limits dialog (3.1.3.2.1) change this so that all, none, or only selected spectra are automatically replotted.

3.1.1.1 Mouse functions on the graphics screen

Several mouse functions are available in the graphics screen. Normal left-clicking does nothing, so you can click anywhere in the screen or title bar to bring the window into focus after working in another window.

Right-clicking at a point on the screen causes the corresponding wavelength and absorbance to be displayed in a textbox at the bottom left of the window(3.1.2.1). If you hold the right button down as you move the mouse, the values in the box are continually updated to the current point.

Middle-clicking (i.e. scrollwheel) at a point on or near a spectrum causes that spectrum to flash on and off several times, and a message box informs you of the number and comment of that trace. It also causes that trace to be selected in the dashboard list. This is complementary to the "Identify" button on the console.

The algorithm chooses the trace in memory that is closest to (and within a threshold distance of) the clicked point, whether it is displayed or not (this version of the program does not keep track of which traces are displayed). So if traces are very dense in the area clicked, and not all displayed, there is a possibility that another undisplayed trace will be identified instead of the one you intended. But this will be obvious because the identified trace will flash and remain plotted, and you will see that this is not the trace you intended.

Dragging a rectangle, while holding down the left button, brings up the screen limits dialog(3.1.3.2.1), prefilled with the wavelength and absorbance limits you selected with the rectangle. Clicking the upper OK button or hitting <enter> then redraws the screen with these limits. The second row (fullscale, midscale) is set with the current absorbance limits, so clicking the second button only changes the wavelength range. If you only want to change the absorbance scale, hit the "fullscale" button by the wavelength settings before hitting OK or <enter>

3.1.2 Buttons below the graphics screen

Controls at the bottom of the screen:

3.1.2.1. "Drop Files here"- Native format files (binary, .mtx), (old)Shimadzu .spc files, and .csv files can be dragged here to load. See Chapter 4 "Data formats". With the native binary format, it will load only the dropped file, unless you hold

the <ctl> key down (making "+" sign by the mouse cursor), in which case that file and any subsequent file in the series will be loaded. With Shimadzu .spc files, any subsequent files in the series are loaded. With .mtx files or .csv files, which can hold multiple files, all spectra in the file are loaded. The data directory from which the file came will become the new data directory. If there are already spectra loaded and the wavelength range or sampling interval is different, you will get the usual warning and a choice to delete and continue or not.

This field is also used with the right-mouse button to display wavelength and absorbance at cursor positioned on screen, so the "drop files" label may be gone, but it still functions to load files.

3.1.2.2. Status bar- tells how many spectra are loaded and how many spectra of the current size would fit in memory.

3.1.2.3. A label displaying the Current data directory - Where most files will be loaded and saved if no path is given.

3.1.2.4. Exit button - to quit the program. Alternatively ctrl-Q or file:exit

3.1.2.5. Console - Brings up trace selection window (console, dashboard) for listing the spectra, selecting, and many operations on selected spectra.

3.1.2.6. Plot All - Plot all spectra in memory on the screen

3.1.2.7 Clear Screen - Erase spectra and redraw frame.

3.1.2.8. Zoom/Pan buttons- This tool lets you zoom in or out on the absorbance and/or wavelength scale, as well as move the view up and down the scales. Note- Move up or down is ambiguous. The way it is implemented here, move up makes the spectra move up on the screen, i.e. your viewpoint moves down to lower absorbance range. Likewise for wavelength, if that box is ticked, move up causes spectral features to move to the right, your view moves to shorter wavelength. Let me know if you think this should be the other way around.

If the "fine" box is not checked, The UP and Down buttons translate by 1/2 of full scale. Zoom in and out expand or contract the scale by a factor of two for each click. Expansion or contraction is about the midpoint of the scale. If your zero baseline is along the bottom of the screen, zoom in will put it a half-scale below, and up will bring it back to the bottom, at twice the size, so "in then up". Likewise if you zoom out the baseline will come up and the lower quarter of the screen will be empty. If you first press down, putting the baseline 1/2 screen below the bottom, zooming out will bring it back to the bottom. So, "down and out". However if you make large adjustments requiring many clicks, it is easier to first press up to put baseline at the center, Then as many clicks in or out as needed to make the peaks just go offscale on top, then down to return baseline to the bottom.

The zoom and Pan buttons only affect the display. The whole spectrum is retained in memory with unchanged absorbance values and starting and ending wavelength. All spectra in memory at any one time must have the same start and ending wavelength and sampling interval (See “load into” 3.1.3.1.1, if you want to load a spectrum with different WL parameters while keeping the spectra currently

in memory). You can change starting and ending wavelengths of the spectra themselves with "adjust spectral range" under the operations menu. To work with a subset (shorter wavelength range or sparser sampling) of a spectrum or spectra, you can save in .mtx form which allows saving a subset, then reload the .mtx file.

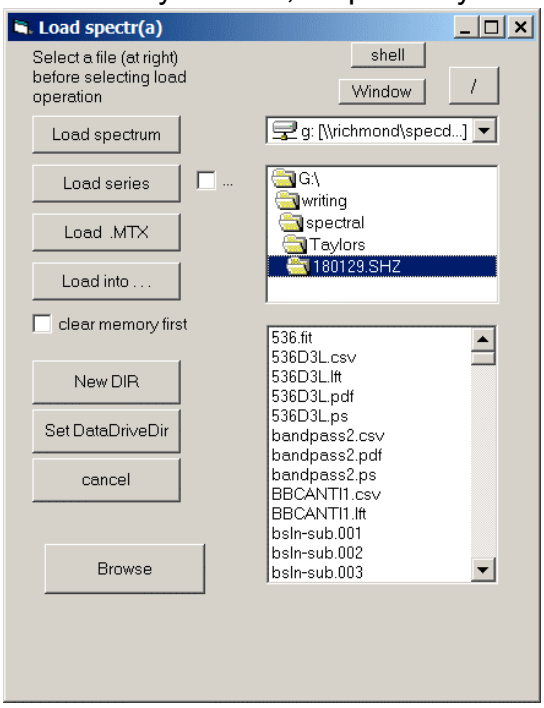
3.1.3 Dropdown menus on the graphic window

3.1.3.1 File Menu:

3.1.3.1.1 Open (Load) (Kbd shortcut <Ctrl-l>)**-** This brings up an old-style file-selection dialog with a number of options. In addition to selecting files to load, it allows you to change the current data directory, create new directories, and open an explorer window or command shell in a given directory.

(For information about the different formats that can be loaded, see section x. For more information on these old-style file-open dialog boxes, see appendix)

A drop down menu and two scrollable list boxes allow you to select the drive, directory and file; respectively. Then:



Load spectrum- loads the selected spectrum if it is in the native binary format.

Load series- If the selected spectrum has a 3-digit numeric extension, that spectrum will be loaded, the extension will be incremented and the next file in the series will be loaded; repeated until the next file doesn't exist or memory is full.

If the box to the right is checked, a second dialog allows you to specify the first and last spectra to load.

Load .mtx- Load the file as native ASCII format (normally .mat or .mtx).

load into- One restriction of the program is that all spectra loaded into the active traces must have the same wavelength range and sampling interval. Load into allows you to select a spectrum with a different but (partially) overlapping wavelength range. It will be truncated and extended with zero's in order to fit the current parameters.

You must select a file *before* pressing one of the load buttons. The file list shows all files in the dir without respect for extension.

The buttons labeled *shell* and *window*, above the drive selection menu, open a command prompt or an explorer window with the currently selected directory without changing current directory of scanedit.

new dir- creates a directory in the currently selected dir and changes to that in the directory selection list.

set datadrive, dir - sets the currently selected drive and directory as the default for loading or saving files if you do not specify a path; and closes the dialog

cancel- closes the dialog without loading anything or changing the data directory.

Browse - brings up a standard windows file selection box to choose the spectrum, then returns to the dialog described above for selection of file type.

The file type filter can be set for "all files", binary spectrum files, or ascii .mtx files) Note you must select a file in order for browse to return any information. On return that file will be selected in the old-style selection dialog. To load the file, click the appropriate button. If you hit "set current . . .", "make directory", "open window" or "shell", the file will be ignored and only the directory information will be used.

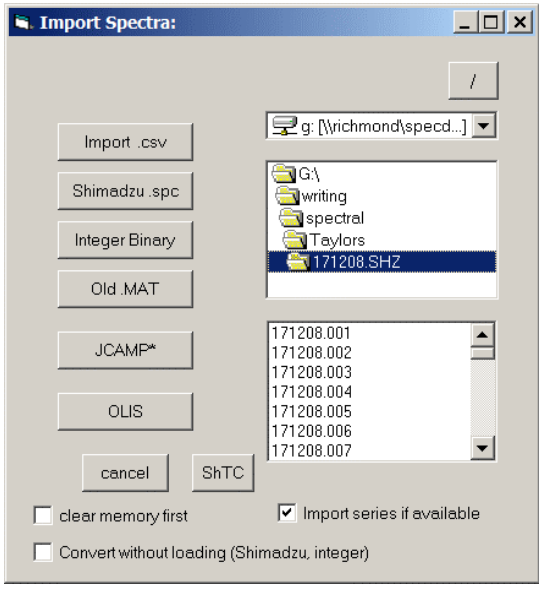
This **file:open** item is your main tool for navigating the filesystem to find files or set the current data directory (more or less equal to the current directory). You can navigate using the old-style disk and directory list boxes on the form, or click "browse" to use a more modern file selection dialog. If you navigate to and load a file, the current directory will change to that file's directory. If you want to change directory without loading anything, navigate to the directory and hit "set current . . ." on the form. You can also set the directory at startup by dragging a file from that directory onto the shortcut to scanedit, or after startup by dragging it to the "drop files here" box (3.1.2.1). If the file is not a spectrum, you get a warning and the option to cancel, but by this time the directory has already been changed and will be used for saving binary files and "File:open" will initially show the contents of the new directory.

3.1.3.1.2. Save- Opens the file-selection gui (console, dashboard) for you to choose a spectrum in memory and save it. Don't use it- just open the console directly and use the save: binary, .mtx, and .csv buttons on the dashboard (3.2.2.15)

3.1.3.1.3. Save Ascii- Same but for the ascii (.mat, .mtx) native format. Again just go to the dashboard and use the save .mtx button.

3.1.3.1.4. Save All- Save everything in memory as binary files in a series with specified name and starting index. Spectra are saved in native binary format, in a series with the filename and the starting filenumber you specify; in the order they are present in memory. This allows the common operation of saving everything in memory without having to go to the dashboard, select all, and save binary.

3.1.3.1.5. Import (Kbd shortcut <Ctrl-I>)- This brings up a dialog much like the load dialog above, but allows loading other formats. Details are provided in the Chapter 4, "Spectral formats and Import".



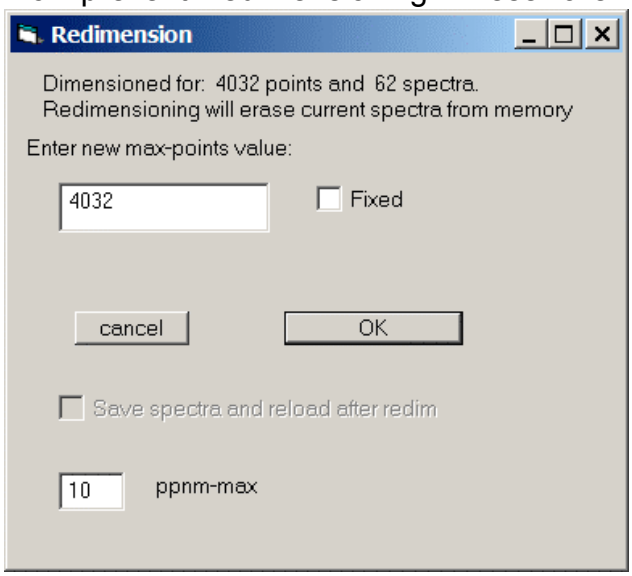
"Import .csv" does just that, but the .csv file should be prepared as described in Chapter 4. *"Shimadzu"* refers to spectra saved as .spc files by their older UV-PC program. If the file is one of series as described in 4.2.1, following files in the series will also be loaded. "Integer binary" and *"Old .mat"* are mainly for the developer's lab, referring to formats saved by a previous version of the program. JCAMP import is described in 4.2.3. OLIS import is still in the testing stage, if you have trouble with them (or anything else) we would like to hear about it.

"Clear memory first" Sets the number of traces in memory to zero (traces are not erased and can be recovered, if not overwritten, by generating a spectrum at a higher trace #). "Import series" causes Shimadzu .spc files to be loaded as a series if the last two digits of the 8-character filename form consecutive numbers (myspec01.spc, myspec02.spc ...) *"Convert wo loading"* causes Shimadzu .spc files to be converted and saved immediately, with the same basename as the .spc files. This ensures that the converted spectra have not been manipulated before saving.

3.1.3.1.6. Clear Memory. Erase all traces from memory. Not used much, since the load and import menus give the option to clear memory first, and loading spectra with different wavelength range clears memory in any case. The traces are not actually cleared, and can be recovered by generating a spectrum (Lin COMB) in a trace number above those to be recovered.

3.1.3.1.7. Print Comments- Writes a list of the comments of all spectra in memory to a text file "comments.txt" in the current directory, and opens the file in MS NotePad. This gets around the limitation of the trace list on dashboard to only 35 to 50 spectra due to lack of a scroll bar. It also presents the comment in a form that can be copied and pasted into other documents, which is not possible from the dashboard.

3.1.3.1.8 Redimension- Normally redimensioning the data array is automatic and need not concern the user. But there are a few instances where it is necessary to control the dimension or prevent redimensioning These are described in (load into 3.1.2.1.1), (resampling .4.Notes),



The program limits the total number of datapoints in the data array, that is the number of spectra times the number of points in each spectrum to about 250,000. When new spectra are loaded, the number of points is set to accommodate it, and the number of spectra is set to the maximum that can be accommodated with that number of points.

Text at the top of the re-dimension dialog gives the number of points and the number of spectra the array is currently dimensioned for. To redimension, you change the number of points (in the first text box). If you also check the box labeled "Fixed", it prevents automatic redimensioning when new spectra are loaded.

Internally, wavelength is represented by an integer, which is related to actual wavelength by another integer, ppnm (points per nm), which by default is 10. The sampling interval, in terms of internal wavelength units is another integer. So the sampling interval must be a multiple of 0.1 nm. If you load a .mtz file or .csv file with 4, 8, or 20 points per nm, the ppnm is automatically set to 8; or to 20 in the latter case, and everything works fine. Unfortunately the current native

binary format for saving spectra does not save the value of ppnm, only the starting wavelength and internal sampling interval. Therefore if you load a binary file with incompatible sampling interval you will notice the wavelength scale is wrong. You can use the redimension dialog to change ppnm to correct it (see resampling (4.2.2.4). Changing ppnm, or checking the "fixed" box, does not trigger redimension, so any spectra in memory are preserved. Changing the number of points redimensions the array, and any spectra in memory are lost.

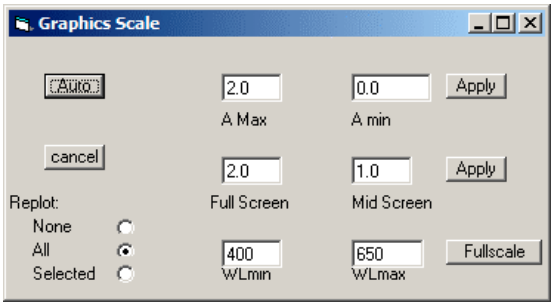
3.1.3.1.8. Exit- standard file:exit, quits the program. Any traces in memory will be lost. Keyboard shortcut Ctrl-Q>, or there is an "Exit" button (3.1.2.4)

3.1.3.2 Display menu

3.1.3.2.1 Set Limits (kbd shortcut <ctrl-L>)- Adjusts the wavelength and absorbance ranges displayed.

There are several methods by which the screen display limits can be set in order to zoom in on small features or zoom out to display the entire spectrum. These are: the screen limits dialog box (menu display:set-screen-limits or <ctrl-L>), the zoom in/out/up/down buttons at lower right of screen window, and ctrl-dragging a rectangle around the desired region. Each is discussed individually below. Autoscaling, to optimally display selected spectra over a given WL range, is available in the “console” window, discussed in that section. Other methods of setting the display limits are discussed in Chapter 6.

With this dialog the Absorbance scale can be set in two ways: the upper pair of textboxes set max and min values of absorbance displayed, while the middle pair set absorbance range fullscale and value at center (sometimes it is useful if you know the absorbance of a feature, to zoom in around it, or for looking at difference spectra, which extend equally negative and positive of zero). When the form is invoked from the menu or <ctrl-L>, both are set to the current display settings. Changing any of the four boxes sets the corresponding “apply” button to default so <enter> enters the values you entered. Otherwise clicking on either “apply” button applies the settings in boxes to its left).



The bottom two textboxes are for setting the wavelength range. The button to their right (fullscale) sets them to the current wavelength range of the current spectra. If only the wavelength range is being changed, click either of the apply buttons to apply the new wavelength range while keeping the original absorbance range.

Clicking the cancel button, or hitting escape, dismisses the GUI with no changes made.

Changing the scale clears the screen. The radio buttons under "Replot" determine whether and which traces will be automatically redrawn at the new scale. By default all traces in memory are redrawn. By choosing "selected" only the traces selected (on the dashboard) are re-plotted, and by choosing "None" no traces are replotted allowing you to replot the ones you want manually). The

program currently does not keep track of which traces have been drawn, so there is no way to redraw only those that were previously drawn.

The "auto" button sets min and max absorbance to the minimum and maximum absorbance of the traces currently in memory. Only the points between the min and max wavelength are considered. If you are changing the wavelength range at the same time, the new wavelength range is used. Thus if there is very high end absorbance at short wavelength which you want to ignore in setting the scale to view long-wavelength features, you can set the wavelength you want for setting the scale before clicking Auto. Then invoke the gui a second time and click fullscale, apply if you want the full wavelength range nonetheless.

The Auto button is default when the gui comes up, and <ctrl-L> is shortcut for opening the gui, so a quick way to set a reasonable scale after loading a new set of spectra is <ctrl-L><enter>.

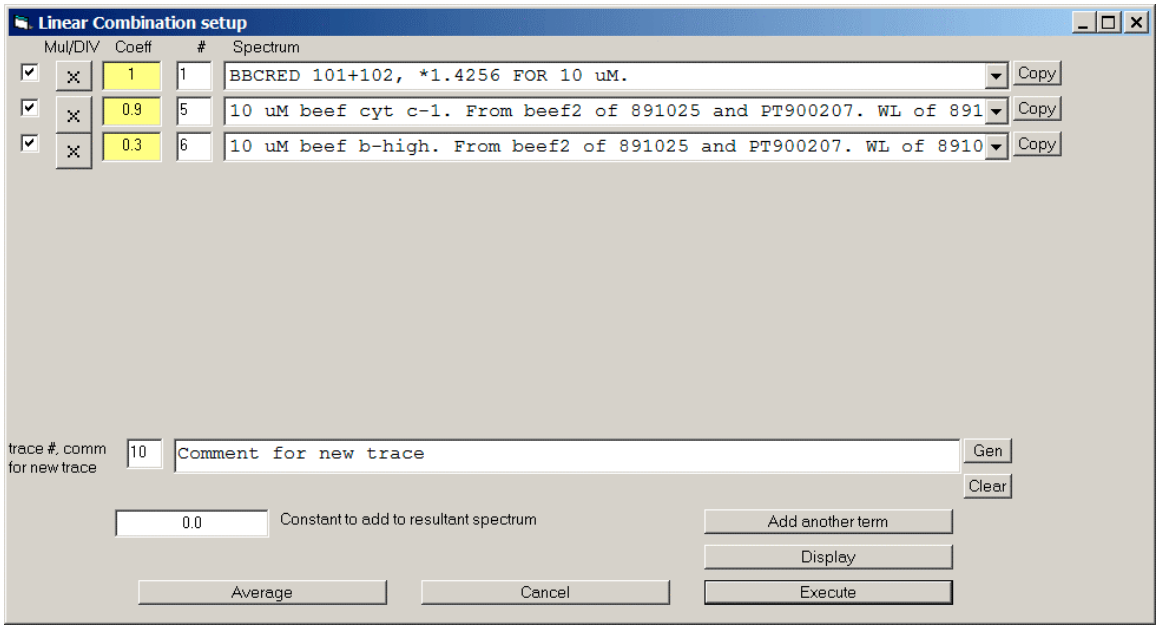
Note that clicking “fullscale” only puts the WL values in the textbox, leaving the gui for you to make further changes if desired before clicking “Apply”. On the other hand hitting “Auto” dismisses the GUI and applies the calculated settings immediately.

If there are no traces in memory, hitting “Auto” dismisses the gui but does not change the scale. If for some reason you want to change the scale when there are no traces in memory, you need to enter the values and click “Apply”

Other ways to set the display scale limits are discussed in chapter 6, and sections 3.1.1 and 3.1.2.9

3.1.3.2.2. Clear screen - redundat with button of the same name (3.1.2.6)
3.1.3.2.3. Plot all - also redundant. Plots everything in memory
3.1.3.2.4 plot. . . - enter a range of spectra to plot on the screen. Useful if you have too many spectra for the tacelist and you want to clear the screen and plot some that you cannot select in tracelist.
3.1.3.5. Resize. Use this to redraw the graphics screen and controls if you have changed the size of the graphics window by dragging or maximizing. (This can probably be made automatic, triggered by windows' resize event - look into it for next edition.)

3.1.3.3 “Operations” menu
3.1.3.3.1 Linear Combination
Add up to 10 spectra, each scaled by a different factor, and optionally add a constant offset to the result. By using a negative coefficient, spectra can be subtracted. In the simplest case, a single spectrum can be multiplied by a constant. (In this case, and especially if the same multiplier/divisor is to be applied to more than one spectrum, it is easier to use the console "divide/multiply" button.
When you select linear combination, it raises a form, shown in Figure 3.1 for selecting the spectra (from among those in memory) and specifying the multiplier for that spectrum. The line at the top is for selecting the first spectrum and setting its multiplier. Clicking "Add another term" presents a similar line for the second spectrum, and so on up to 10.
Figure 3.1 depicts the form after three spectra have been selected. The longest field here displays the comment of the selected spectra, and clicking the arrow to the right of the box lets you select from a list of all spectra in memory. The check-



box on the left indicates the spectrum is to be used. (It is automatically checked and actually has no use unless you want to omit one spectrum after selecting it) Next, there is a button with a Times Sign (×). Clicking this (or tabbing to it and hitting spacebar) toggles it to a divide-by sign.

The next box is the multiplier, preset to 1.0 and highlighted in yellow (since you likely want to change this) Next is a text box with the trace number of the spectrum. This is set by selecting the spectrum in the next box. However this number indicates the spectrum that will be used, so if you remember the trace number you could enter it here instead. Then hit <tab> and the comment will be updated.

The next line after the selected spectra has the number of the trace you want the new spectrum to go in (defaults to the next available trace, it is ok if you want to save it over (in the same trace as) one of the original spectra), and a text box to enter the comment for the new trace. Two buttons to the right of this, clear remove any text, Gen generates a comment based on the trace numbers and coefficients. If you click the "Copy" button to the right of one of the input spectra, it copies that comment and appends to the comment for new trace.

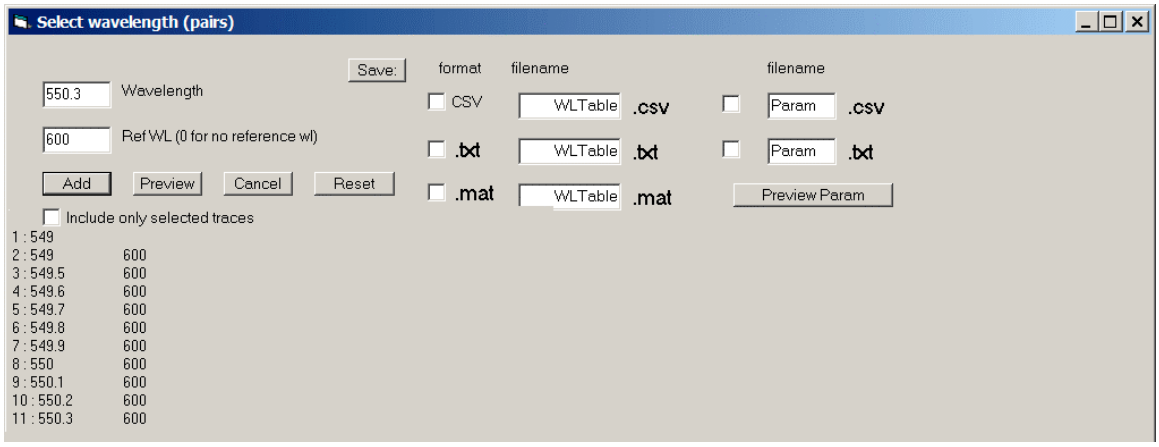
The "Display" buttons lets you preview the Linear Combination then return to the dialog, so you can adjust the coefficients by trial and error without starting over for each trial. "Average" sets all the coefficients to 1/N to make the average (but see average button on the Console). Finally "Execute" calculates the new spectrum and closes the dialog.

For a faster set-up, select the spectra in the tracelist and activate "Set up Lin Comb" (3.2.3.3) in the "other functions" dropdown menu. This brings up the Lin Comb dialog prefilled, just enter the desired coefficients.

3.1.3.3.2 Print table: save a text or spreadsheet listing absorbance at selected wavelengths.

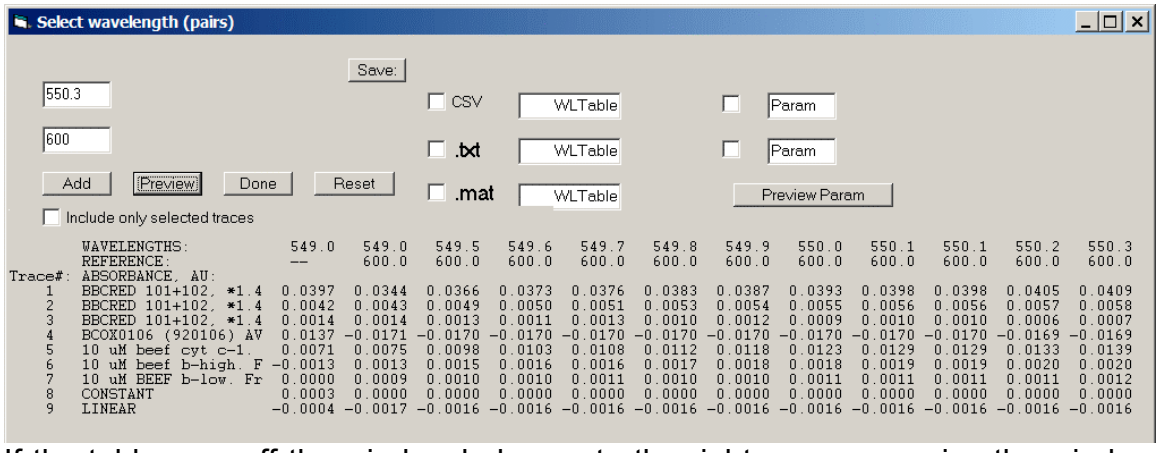
Enter a list of wavelengths, or wavelength pairs, and print a table of absorbance values at those wavelengths. For each point, enter the wavelength in the indicated box. If you want a reference wavelength to be subtracted out, enter in "ref. WL", otherwise leave blank or zero. Click "Add" (or hit enter) to enter it in the list. Repeat until all wavelengths have been entered.

Figure 3.2a



If you want to include only spectra that have been selected in the trace-select window, click that box. Then click preview to print the table here on the form.

Figure 3.2b



If the table runs off the window below or to the right, you can resize the window and hit “preview” again. When you have all the wavelengths selected, check boxes for the output(s) you want: text, comma-separated values (for spreadsheet), or .mat . (This is *not* the .mat/.mtx file for loading spectra into scanedit).

The current binary file format includes 5 "parameters" with each spectrum. In the case of spectra derived from .spc files, these include information about spectral bandpass, scan rate and sampling density. Clicking "Preview Param" lets you see the parameters.

3.1.3.3.3 Stack spectra - Constants are added to each spectrum to displace it so that it is stacked above (or below) the previous one as close as possible without overlap. The minimum separation to avoid overlap is set by the user (A value of zero lets the spectra touch (overlap) in just one point), as is the direction: first on top, or first on bottom (default). This is useful for comparing the shape of spectra when the absolute value is less important, and for seeing trends in the changes in a series of spectra (e.g. time course, elution profile, titration).

3.1.3.3.4 Differentiate - The first, second, and third derivatives of the spectrum at each point are calculated. This is done as in the "smooth" function below, but at each point the derivatives of the polynomial are calculated and saved as the derivatives of the spectrum at that point. Input is the number of the spectrum to

be differentiated and the width of the sliding window. The output spectra are placed in the next 4 available traces. See Chapter 10 for details.

3.1.3.3.5 More derivatives- The same as "differentiate" but allowing to calculate derivatives beyond the third.

3.1.3.3.6 Integrate - A cumulative integral is calculated, starting with zero at the leftmost point, and put into a new spectrum. This is a spectrum whose derivative is the original spectrum. The average value of the spectrum can be subtracted before integration, In which case the integral spectrum will start and end at zero.

3.1.3.3.7 Smooth - spectra are smoothed by polynomial fit. For each point, the original data in a window centered on that point is fit to a cubic polynomial, and the value of that polynomial at the central point is taken as the new value at that point. Points within half the width of the window of either end of the spectrum are evaluated from the polynomial fit to the first and last window-width points. The number of points in the window is set by the user- higher values give more smoothing but perhaps some distortion of the spectra, lower values give less smoothing but more faithful reproduction. Values corresponding to 1.5 - 2.1 nm seem to be appropriate for UV-vis spectra. The number of points in the window must be greater than 4, since we are solving for four coefficients of the polynomial. See Chapter 10 for details.

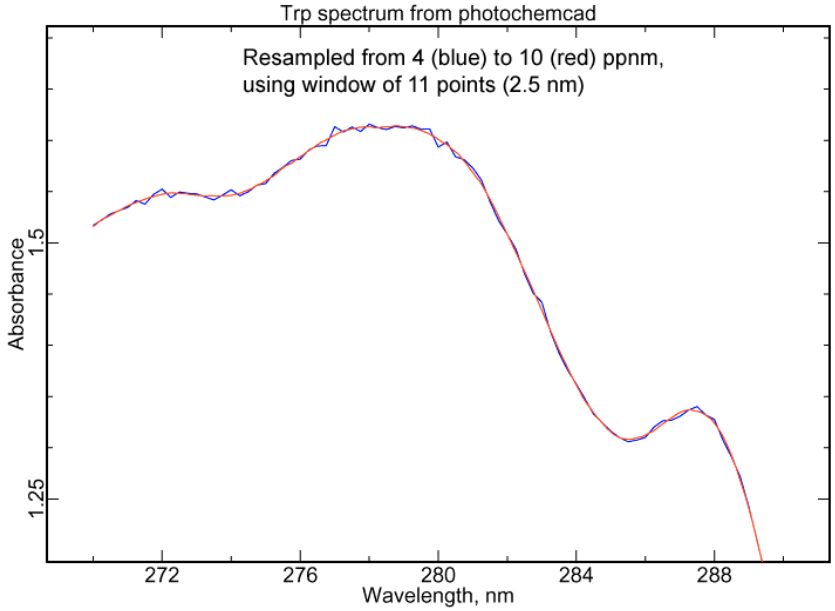
3.1.3.3.8 Shift Spectra - Shift spectra to longer or shorter wavelengths, by even multiple of the sampling interval. The current wavelength range is retained, points shifting out on one end are lost and points on the other end are filled with zero. To keep all points and re-define the wavelength range, you can save as .mtx and edit the .mtx file to change the starting wavelength (see format of mtx files, appendix)

3.1.3.3.9 Add Baseline Polynomials- Generate one, two, or three spectra to be used in fitting any uncorrected baseline in the lsq procedure, and put them in the next available traces. The supplemented set of spectra needs to be saved as a .mtx file and used in fitting (7.2, 7.5).

3.1.3.3.10 Adjust spectral Range - Restrict (or extend) the wavelength range to what is currently displayed, with the current sampling density. First use "Display:Set Limits" to select the display wavelength range, then activate this command. All spectra in memory will be truncated/extended to that range and can be saved as such. In case of extended range, regions that do not exist in the original spectra are set to zero absorbance.

3.1.3.3.11 Resample - interpolates between points to generate a spectrum on a different (finer or coarser) grid. The result is saved in a .mtx file. This does not just interpolate linearly between the points, it fits the old spectrum in a sliding window about each old point to a cubic function, and evaluates that function at the new point(s) nearest that old point. This avoids flattening curves ("polygonization"), and results in significant smoothing of noise in the original spectrum, as seen in the figure. The amount of smoothing increases with the

width of the window. The input spectrum must be sampled at constant intervals and should be loaded already. Parameters are the trace number of the input spectrum, the new sampling interval and the width of the sliding window. See Chapter 10 for more advice.



3.1.3.3.12 Console - Brings up the "console" or "dashboard" window, same as pressing the "console" button below. <ctrl-D> is a shortcut.

3.1.3.4 SVD menu

3.1.3.4.1 SVD - Performs Singular Value Decomposition on a series of spectra. The spectra should be in native binary format and form a contiguous series, but should not be loaded into memory. Enter the directory, basename and first and last numbers of the spectra to be decomposed, or use the "Browse" and "Last" buttons to select them. Also put a filename prefix for the files to be generated, or leave the default "abc". Then click "Begin". For more details see Chapter 7. "SVD".

The procedure will go through several stages, with some explanation of what it is doing, then it will perform the two tests described below. The singular spectra will be saved in a series with basename <prefix>SV; the squared singular values in a text file <prefix>.evl, and the right singular vectors ("singular timecourses") weighted by their singular values are in a <prefix>.csv spreadsheet with the vectors as columns. See chapter 7.

3.1.3.4.2 Test EVD - test the eigenvalue decomposition performed as 1 step of the singular value decomposition above - for debugging. No real use except confidence-building.

3.1.3.4.3 Reconstruct Orig. Spectra with N Components - Tests the SVD operation by regenerating the original spectra from the singular spectra and vectors. By default all singular spectra with non-zero singular values are used, and the spectra should be regenerated within the precision of the arithmetic, RMSD 10^{-10} or smaller. To see how many of the singular spectra are required to reproduce the experimental spectra within noise level, you can use a smaller

number. Using limited number N of Singular spectra, tells what portion of the variance (from zero) is explained by the N largest Singular spectra.

3.1.3.4.4 Real spectra from fit to Singular timecourses - see details Chapter 7. Given spectr(a) expressed as a linear combination of Singular spectra, generate the real absorbance spectra. This is useful when you have performed a global fit of the singular timecourses to some function such as exponential decay, Gaussian elution peaks, or Nernst or Hasselbach equation, and you want to generate the spectra of the resulting components.

3.1.3.5 Fourier menu
Calculate Fourier transform, invert the transform with series truncation or damping, resample a spectrum to different sampling interval. For details, see chapter 9, Fourier Transform.
3.1.3.5.1 Fourier Transform - calculates the discrete Fourier transform of a spectrum in memory, saving it in the next available trace.
3.1.3.5.2 Inv FT - Calculates the inverse transform, regenerating the original spectrum, optionally with truncation of the Fourier series at a specified frequency.
3.1.3.5.3 Damp a transform - apply damping factor to the terms of a Fourier transformed spectrum
3.1.3.5.4 Damped inverse FT - carry out inverse transform to regenerate the spectrum, applying damping factors calculated from the formula damping factor = 1/(1+Fs/F) where F is the frequency of the Fourier term and Fs is the "shoulder" frequency at which the damping factor is 1/2. You specify the shoulder frequency.
3.1.3.5.5 Frequency Spectrum - Squares every term of the Fourier Transform, making a power spectrum on the same scale as the transform
3.1.3.5.6 Draw Fourier (Frequency) axis - The wavelength scale is replaced by a frequency scale. In full scale this goes from 0 to one less than the number of points. The units are cycles/nm if the spectral scale is nm.
3.1.3.5.7 Inv. F.T. with resampling - An alternative to the resample procedure under the "operations" menu. The inverse Fourier transform is evaluated at the wavelength points required for the new sampling
3.1.3.5.8 Fourier components- Each Fourier term is plotted and the cumulative sum at each step is shown. Optionally save the Fourier components and the incremental sums to two series of spectra.

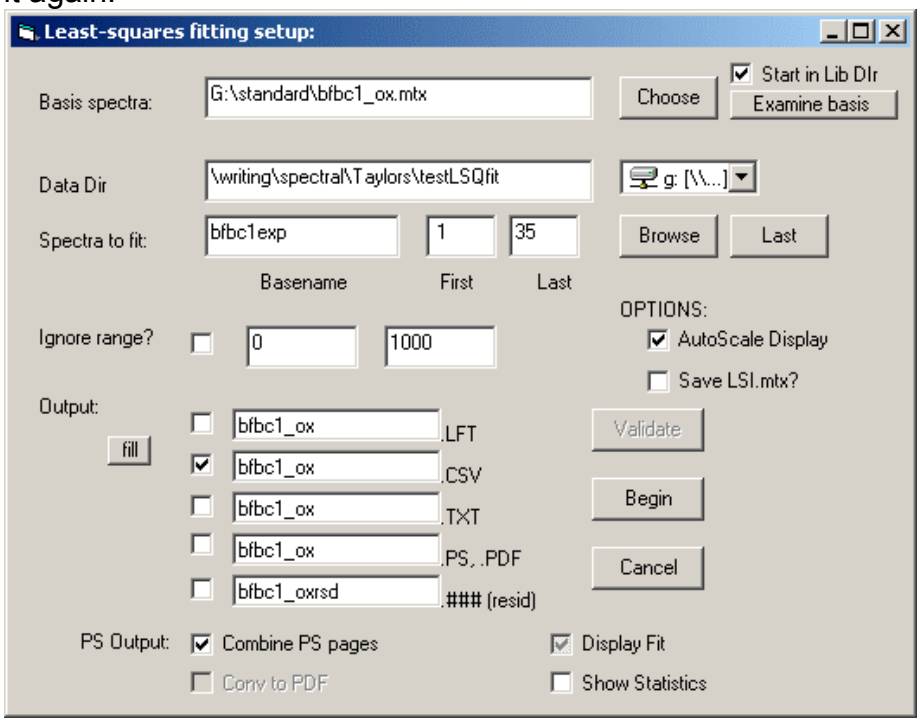
3.1.3.6 LSQ fit menu - The only item is LSQ-Fit: Should put "superimpose spectra" here instead of on tracelist.

This does Classic Least-squares fitting of a spectrum to a linear combination of component spectra, for analyzing spectra of mixtures. Selecting LSQ-fit brings up the dialog for fitting one or a series of experimental spectra (in binary format) to a linear combination of standard spectra present in a .mtx file.

The first line of controls, labeled "Basis Spectra" concerns the fitting spectra. These could be, for example, spectra, in pure form, of all the substances that may be found in the samples being analyzed. They should be saved in a single .mtx file (4.1.2). Click the choose button to find your standard .mtx file for the analysis. If the "Start in Lib Dir" button is checked, the initial directory will be

set to your spec-lib directory (see 5.5). Otherwise it will start in the current directory.

If you want to look at the basis spectra, click "examine basis spectra". The form will be hidden and the basis spectra will be loaded into memory and plotted. However anything that was entered on the form will be retained when you invoke it again.



The next two lines specify the spectra being fit. They must be saved in binary format, in a series with consecutive numbers. Their spectral range must include that of the fitting spectra, but can extend farther in either direction. Click the "browse" button to locate the first in the series. Enter the number of the last (or click last to locate it). It is OK for the last to be the same as the first, in case only one spectrum is being analyzed.

The "ignore range" boxes allow you ignore part of the spectral range of the fitting spectra. This is useful, for example, if a peak has gone above the linear range of the spectrophotometer, or if some unknown contaminant interferes with the fit in part of the range. The boxes come prefilled with 0 to 1000, which would normally ignore all if the checkbox were ticked. However it is useful for ignoring a range on one end or the other- if you want to exclude wavelength below 400 you might put 399.9 in the second box; or to exclude all above 600, put 600.1 in the first box. The check box is automatically ticked if you change either box. Uncheck it if you want to go back to using the full range.

Ignoring part of the range does not allow you to analyze spectra that do not cover the entire range of the basis (fitting) spectra. Rather you should save the fitting spectra in a new .mtx file with narrower range (3.2.2.14).

The next five lines allow you to select one or more of five different types of output, and specify a filename for each. The button labeled "Fill" copies the basename of the .mtx being used into the filename for each of these, and a checkbox by each determines whether that output will be produced. The idea is that you will be working in a directory created for the dataset being analyzed, and you may be analyzing it with several different .mtx files. This will connect the

output files to the .mtx with which they were created. But you are free to enter whatever you want for the filenames.

The .csv file is the most generally useful output, so it is checked by default. This gives a comma-separated-values file that can be opened in a spreadsheet program like MS Excel to give the results in tabular form, ready for plotting or further calculations. After the first, every row corresponds to one of the fitted spectra. The first column is just the file-number of the spectrum in the series being fit. The second is the descriptive comment of that spectrum, The next N columns, where N is the number of fitting spectra, contain the amounts of each of the fitting spectra to best fit that spectrum. The final two columns contain the RMS magnitude of that spectrum and the RMS error of the fit. The top row has in the second field, the file-path of the .mtx fitting spectra. The next N fields contain the descriptive comments of each of the fitting spectra.

The .txt file is similar to the .csv output but with tabs instead of commas, to make a plain-text table of the results. The descriptive comments are truncated to the first 20 characters.

The .lft file is a hangover of the original output format; it is kept only because it is used as input for some other programs. Actually it is produced whether its box is checked or not.

The box labeled ".PS, .PDF" controls whether a postscript file with the results is created. If "convert to pdf" is checked at the bottom of the form (and if scanedit can find GhostScript installed on your computer- (see 5.7), a pdf file will also be made. These files have one page for each fitted spectrum, displaying the original spectrum, the best fit, and the residual trace; in blue, green, and red, respectively. Because of the vector graphics and the ability to show pdf at magnification or print it to a 600 dpi printer, this will give a higher resolution view, useful for distinguishing the best fit from the original spectrum when they are nearly identical.

The box labeled "rsd.###" enables filing the residuals for each fitted spectrum in a series with basename ending in "rsd". This can be useful when the residual is significantly non-random, for tracking down the source. For example, by taking representative residual spectra and including in the set of fitting spectra.

The "autoscale" box determines whether the absorbance scale is optimized individually for displaying each spectrum being fit. If not checked, it will use the scale in effect when you start. By default, the "autoscale" box is checked. This is convenient if you are fitting a number of spectra of different magnitudes: The plot showing the fit (and the plot showing the components of the fit, if requested) will be displayed on a scale suitable for the spectrum being fit. The scale is adjusted for the whole spectrum, even if part of the spectrum is being excluded in fitting. If you prefer to set the scale before starting, and not have it changed, uncheck the box. This is good for fitting a single spectrum or a series of spectra of nearly the same scale. In either case, if you select to have a .pdf of the fitting results made, they will be on the same scale as the displayed fit. For more discussion of and advice for least-squares fitting, see chapter 7.

After each fit is displayed, a dialog comes up pre-filled with "Y". If you simply hit enter, it goes on to start the next fit, then waits again for you to inspect the result. If you enter "x" before hitting enter, it goes through the rest of the fits,

displaying each for about 4 seconds but not waiting for further input from the user.

If you click cancel, hit escape, or hit backspace to clear the input box before hitting enter, it proceeds through the rest of the fits without waiting at all. This is pretty fast, but if you really want it to quit without finishing, enter "stop", the output files are closed as is and the procedure exits.

If you type "f" and <enter>, it displays the individual components of the current fit. The tracelist box is set up as a color-coded key, with the comments giving the amount and description of each component. It then asks for a file basename to save the fitting components as binary spectra which you can use in making a decomposition figure. You can just hit enter to skip saving them, if you only wanted to view on screen.

"Display Fit" cannot be unchecked, but as described above you can abort after the first fit. "Show statistics" causes the program to pause after printing some statistics on the screen so you can take note of them. The statistics include cross-correlation and linear independence of the fitting spectra, and "standard deviation multipliers", described in Chapter 7.8. If you check this box it is possible to not select spectra to be fit, and the procedure will return after displaying the statistics. This allows you to test the fitting spectra without actually fitting anything, and any spectra in memory before starting will still be intact. Otherwise, if the box is not checked, you have to select both basis spectra and spectra to be fit or the routine will not proceed.

3.1.3.7 Disk-based - This menu includes several operations that act on spectra in files on disk, rather than spectra loaded into memory. Actually the LSQ-fitting and SVD routines belong to this category, but have their own menus..

3.1.3.7.1 Make PS Figure- lets you plot a series of spectra from the disk, which would be useful if you have more spectra than you can select with trace-select (but it is questionable whether a figure with more than 35 spectra would be interpretable). It also allows some options that have not yet been incorporated in "make figure" on the console, like selecting the color and or dash pattern for each spectrum.

3.1.3.7.2 Subtract mean or best-fit line from spectra - mask baseline changes in order to emphasize differences in shape of the spectra. This routine also lets you take a subset (shorter wavelength range, sparser sampling) of the spectra and save in a new series. Either or both operations can be carried out.

3.1.3.7.3 LSinverse - calculates the generalized, or "least squares" inverse of a matrix

3.1.3.7.4 Spectra × matrix - Generate linear combinations of the spectra in a series, defined by an input matrix.

3.1.3.7.5 Run Script - One disadvantage of the Gui is the inability to feed the program a list of commands to be carried out non-interactively. This will allow it to do just that, for the commands we have implemented so far. More on this in the next edition.

3.1.3.7.6 Average - Allows you to calculate the average of all spectra in a series, where the number of spectra may be too many to select in the trace-list. This could be preparatory to subtracting the average from all, in order to emphasize the changes rather than the absolute spectra (which should be another option here).

3.1.3.7.7 - other items are in development or principally for the developer's lab. To come:

"Right" LSInverse matrix (the above makes a left inverse)

"Matrix × Spectra" where the matrix is on the left.

3.1.3.8 Preferences - See Chapter 5, Customizing ScanEdit

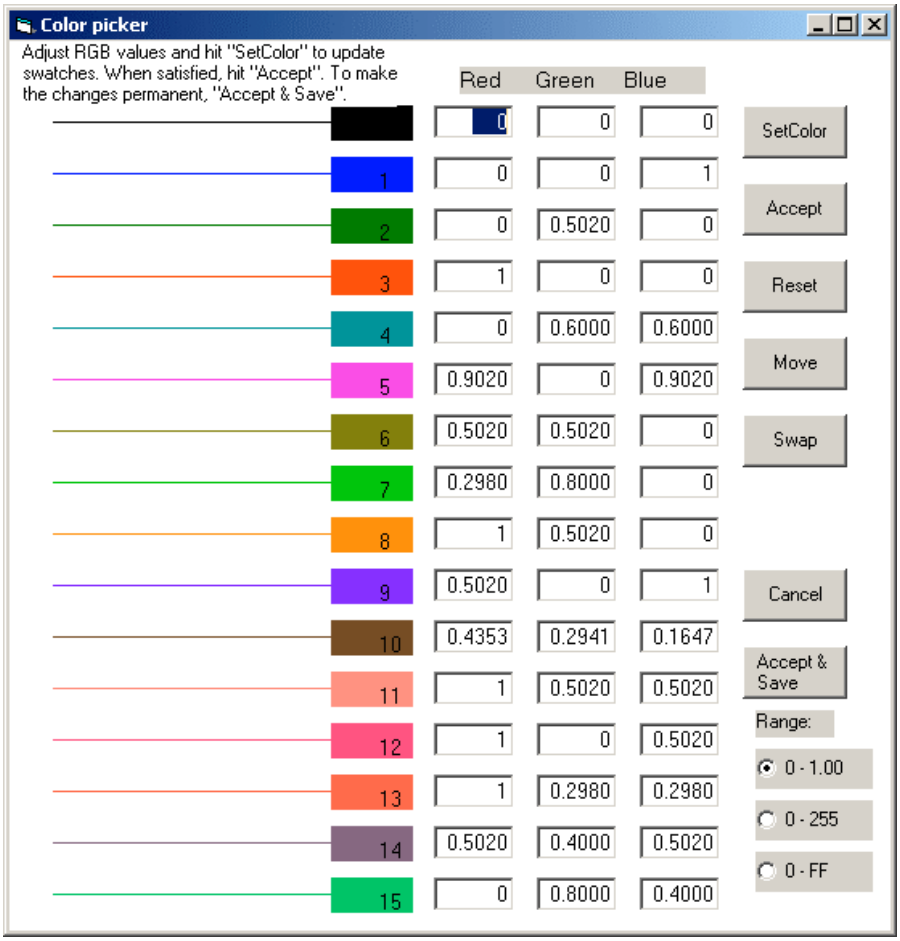
3.1.3.8.1 change linemode - select drawing mode for spectra: dots or lines between points. By default lines. If you select "Auto", lines will be drawn when the points are separated by more than one pixel on the wavelength scale. This can still leave gaps between the points when the curve is nearly vertical.

3.1.3.8.2 Locate GS program - (locate and define ghostscript executable gswin32c.exe for use in converting PS figures to .pdf. See 5.7.

3.1.3.8.3 Set Library directory - set the directory where scanedit looks for standard spectra.

3.1.3.8.4 Edit screen colors - Spectra drawn on the screen cycle through colors 1 through 15 in that order. If you would like to change the palette of colors that scanedit uses for drawing traces on the screen you can use this item. It brings up a form with the current Palette, showing the R,G,B values for each color and drawing a line and swatch in the color, on a black background. You can edit the red, green, and blue color values and click SetColor (or just <tab>, or click in another box) to change each color. By default the range of each RGB value is 0 to 1, but you can use the option buttons to select 0-255 or hex 0-FF. To rearrange the pallet (so less distinguishable colors won't be adjacent) you can use Move (which moves one color to insert it before another, or "Swap" which swaps the positions of two colors. "Reset" restores the colors currently in use. When you are satisfied, click "accept" to copy these into the Palette and close the form. "Accept and save" also saves them to a file in the user's "appdata" directory which will be read the next time ScanEdit is started. Or you can click "Cancel" to leave without change. The color picker is always initiated with what is currently in use, so after accepting you can go back and make further changes even if you have not saved. When you do save, the results are saved in %appdata%/ScanEdit/palette.txt, so if you delete or rename this file you will revert to the program defaults on the next startup.

3.1.3.8.5 Edit PSPlot colors - Does the same but for the palette of colors used in the post-script plot output from "make Figure". Plotting on paper the colors for the traces cycle 1 through 15 then start over at 0 (black). In this case the background for the color swatches and lines is white, letting you see how visible and how distinguishable the traces would be when plotted on white paper. the results are saved in %appdata%/ScanEdit/PSpalette.txt



"Edit PSPlot colors" color-picker dialog. The "Screen Colors" dialog is similar but with a black background so you can see how the traces will look on the screen.

3.1.3.9. About ScanEdit

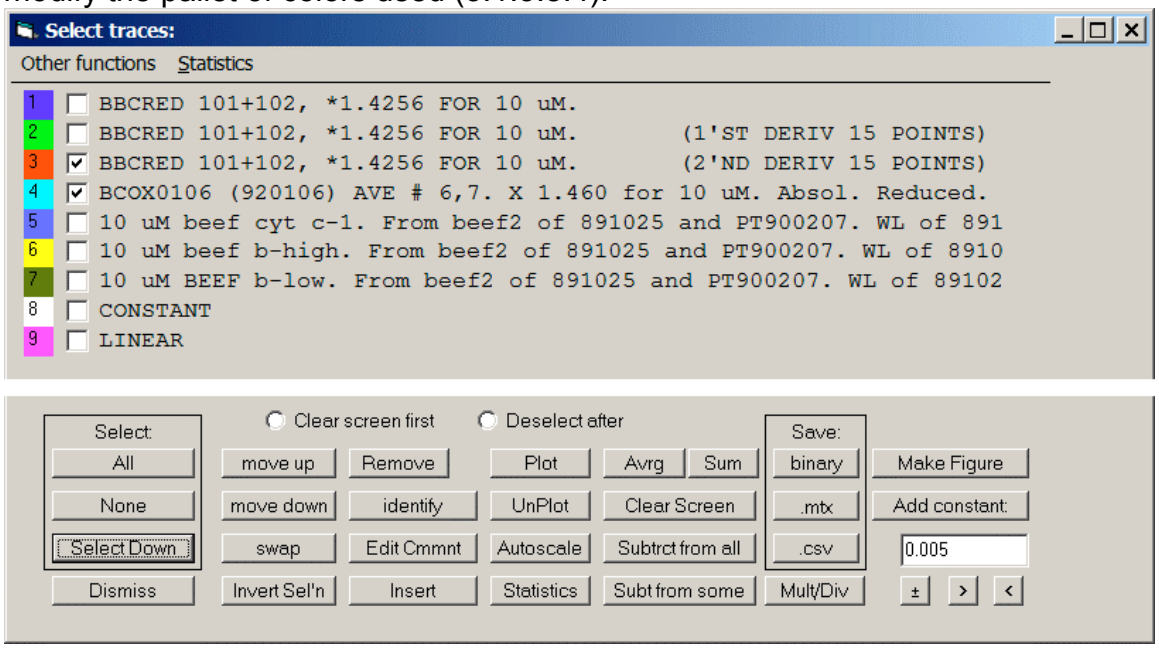
Information about developers, download site, and licensing info (GNU_General_Public_License).

3.2. Trace selection window- ("console", "dashboard")

To see a list of spectra in memory, and for operations that require selecting spectra, open the **console** by clicking the button labeled console on the graphics screen, or with the keyboard shortcut <ctrl-D>. The console comes up to the right of the graphics screen, and of a size that if you have a fairly high resolution screen, both windows will fit.

These are mainly operations that work on selected subgroups of the spectra in memory (the main function of the console being to allow selection of spectra). Activate the console by clicking "**console**" on the lower right of the graphics screen.

3.2.1- Selection List - Up to the first 35 spectra in memory will be listed with their comments and a check box for selecting. Thus a quick way to see what is in memory (e.g. if you just loaded a series or a .mtx file) is to click the console button. The background color around each checkbox in the list matches the color in which the corresponding spectrum is plotted on the screen, providing a key between the list and the spectra. The colors repeat after fifteen. You can modify the pallet of colors used (3.1.3.8.4).



With a moderately high resolution video monitor you can increase the number of spectra listed beyond 35 by stretching the window between top and bottom edges of the screen (be sure to execute "**other functions:resize**", 3.2.3.1), but there is a hard limit to 50 spectra listed at present. If there are more spectra in memory than fit on screen, you can list all by "**file:print comments**" from the graphics screen (3.1.3.1.7), which prints the list of all comments to a file - comments.txt - in the current data directory, and opens it in notepad for you to examine, copy, or save with a new name.

3.2.2.Select - Clicking the checkbox beside the number and comment of each spectrum selects or deselects it in the conventional way. In addition the buttons:

All: select all (up to 35) spectra. <Ctrl>-A is a shortcut.
None: deselect all spectra. <Ctrl-D> is a shortcut

invert selection: unselect what is selected, select the unselected.

Down: propagate the last selection/deselection down to the bottom. Thus to select 5 through 15 (starting with nothing selected), you would click 5 to select it, click **select down** to select everything below 5, click 16 to unselect it, and click **select down** to deselect everything below 16. This rather clumsy-sounding alternative to the conventional click, shift-click to select a range actually works quite well, however the conventional method now works as well for selecting. click an unselected box to select it, then shift -click another box (can be already selected or not), those boxes and all between will be selected.

To select a single spectrum and deselect all others, ctrl-click its box. This avoids the necessity of clicking "none" before selecting a spectrum for the many functions that requires a single spectrum to be selected.

3.2.1.2 Selecting spectra with the dashboard

Bring up the dashboard by clicking the "console" button at lower right of the graphics screen or by <ctrl-D> shortcut. If there are spectra in memory, there will be a column of selection boxes with colors matching the colors of the spectra on the screen. If the spectra have "comments" associated, the comment is displayed to the right of the selection box.

To select a spectrum, check its selection box. Click again to unselect. To select a range, select one spectrum then select another while holding down the shift key. Those two and all spectra between will be selected. To select one spectrum while unselecting all others, control-click its box. The one you click will be selected regardless of its previous state

At the bottom of the console window are buttons to select all, deselect all, invert the selection, or select or deselect down. If you click the "select down" button then the last operation on a spectrum (select or deselect) will be applied to all spectra below that spectrum.

If there are more spectra in memory than can be displayed in the list- The trace list can display a maximum of 50 traces, more like 40 on a monitor with reasonable vertical resolution if the window is resized to max height, and only 35 with its default size. This means you cannot select or operate on any spectra beyond this limit.

3.2.2 - controls at the bottom of the trace selection window.

3.2.2.1 Selection shortcuts - select all. none, down, invert selection

self explanatory except "select down". If you click the "select down" button then the last operation on a spectrum (select or deselect) is repeated for all traces below that one.

3.2.2.2 Dismiss - Hides the selection window without losing the current selections. If you bring it back by the console button or ctrl-D, the same selection boxes will still be checked (even if the spectra in memory have been replaced by new ones). The window can be left open while doing operations in the graphics window, and it will be updated when those operations change the order or number of spectra in memory.

3.2.2.3 Changing the order of spectra- The spectra in memory are listed on the dashboard in the same order they are present in memory, usually the order in which they were loaded or generated. When spectra are saved in a binary series or in a .mtx file, they (or the selected ones of them) are saved in the same order as in memory. If you want to change the order in memory, use the "move up", "move down" buttons after selecting a single spectrum, or "swap" button after selecting two.

Note here “Up” means up in the list, and earlier in memory (smaller trace number). Only the order in memory is changed, the spectra are not. This is distinct from the *Add a constant* button (3.2.2.18) which adds a constant to the selected spectra causing them to move up or down on the screen (without changing the order in the list), and distinct from the up and down buttons in Zoom/Pan (3.1.2.9) which change the display scale limits, causing all spectra to move up or down in the display. Also note that the screen is not automatically redrawn after move or swap, so the color key will be invalid until you replot the spectra)

Up, Down: move a single spectrum up or down in the list and in memory. The selected spectrum is swapped with the one above or below.

Swap – The two selected traces will swap positions in memory.

3.2.2.4 Remove – deletes the selected spectr(a) from memory, and redraws the selection list without them. The screen is not redrawn. (This is "delete" in earlier releases)

3.2.2.5 Identify. The file number in the console-list is background-colored with color used for that spectrum, providing a key to the colors on the screen. Fifteen colors are used sequentially before starting back with the first. In cases with more than 15 spectra, or if the colors are not sufficiently distinct, select the trace of interest in the console list and click the “Identify” button. This briefly toggles the color of the selected spectrum several times to allow its identification on screen. Alternatively, middle-click the trace on the screen (in a place where it is not overlapped by another trace) to get a message-box identifying the trace. Its selection box on the console list will also be checked by doing this.

3.2.2.6 Edit comment – allows you to modify the comment(s) associated with the selected spectr(a). The comments of the selected spectra will be brought up, one after the other, in a text box for editing. Anything beyond 63 characters will be truncated. This is also good for getting the comment in a form where you can select and copy part or all of the comment to the clipboard (the label on the selection list is not copyable). "Cancel" after copying if you don't want to make a change. To copy all the comments, use **FILE:Print Comments** on the graphics window.

3.2.2.7 Plot (on screen) - plots only the selected spectra. if the “clear screen first” option is checked, the screen will be cleared first. This gives the impression of the unselected spectra disappearing.

3.2.2.8 unplot. (plots the selected spectr(a) in black, causing them to disappear. overlapping spectra will have pixels blackened where they overlapped, which is usually not noticeable and can be overcome by re-plotting the mangled spectra.

3.2.2.9 Autoscale works like the *AUTO* button on the *set screen limits* item under the display menu, but only the selected spectra are considered in setting the scale, and only they are replotted after re-scaling the screen. The absorbance display limits will be set so that the selected spectra fill the screen in the selected wavelength range. The wavelength range here is only used in determining the absorbance limits, unless you also check the box "*Also change WL range"*. The WL range is prefilled with the current screen limits rather than the full range of the spectra in memory.

3.2.2.10 Statistics - Calculate statistics on selected spectr(a). Brings up a dialog with three options:

a. Single Trace Functions - Minimum, maximum and mean values, standard deviation, RMS value, norm, RMS deviation from the mean. If you have selected more than one spectrum these statistics are printed for each in turn.

b. Comparing two traces: Mean difference, RMS difference, Correlation (Pearsons), Orthogonality. You must have selected two and only two spectra. Sample output. Traces 1 and 4 were selected for comparison:

```
Comparing two traces, Wavelength range 500.3 - 619.9
 1 BBCRED 101+102, *1.4256 FOR 10 uM.
 4 BCOX0106 (920106) AVE # 6,7. X 1.460 for 10 uM. Absol. Reduced.
 1 Mean: 2.871618E-02       RMS magnitude 3.583146E-02  Norm: 1.239686
 4 Mean: 1.658565E-02       RMS magnitude  0.0175932    Norm: 0.6086839
 1 RMSDev from mean 2.143069E-02
 4 RMSDev from mean 5.868298E-03
RMS difference 2.77882663280029E-02
Dot Product:   0.491504134060961
Cos of angle between  0.6513639
Correlation: -0.5221238
```

Here "correlation" is the Pearsons correlation coefficient, that is from each spectrum the mean was subtracted before calculating. "Cos of angle between" is equal to the correlation coefficient calculated without subtracting the means. Obviously the latter will always be positive if both spectra are absolute, i.e. positive everywhere.

c. *Std Dev vs WL* - with three or more spectra from repeat scans of the same sample (or empty chamber): at each point, take the mean and std deviation of the measurement at the same wavelength in the different spectra. Use these values to make new "spectra" containing the point-wise mean and std deviation as a function of wavelength. This is useful for seeing how the noise varies with wavelength and absorbance level. In case of baseline drift, it may be helpful to first subtract from each spectrum its mean value. We plan to add covariance of the measurement at each wavelength with the adjacent and next-to-adjacent wavelengths, to see how independent the single wavelength measurements are. If the photometer has a time constant commensurate with the sampling rate or slower, correlation is expected. If it is fast and a sample-and-hold is used for acquisition, adjacent wavelengths could be independent.

For the first two options, the statistics will be calculated from the full wavelength range in memory, unless the box "restrict to currently displayed wavelength range" is checked. The output will be written to a text file stats1.txt or stats2.txt and opened in notepad for you to examine, copy, or

save with a new name. The third option has no text output but creates the two new spectra mentioned, in memory.

3.2.2.11 Average and Sum – create a new spectrum which is the average or sum of the selected spectra. A comment is generated for the new spectrum, indicating which traces were averaged or summed, but since these numbers only have relevance to the current situation in memory, you should hit *Edit Comment* and describe what you made if the spectrum is to be saved for later use. Averages can also be generated with *Procedures: Linear Combo* menu item() on the graphics screen, in which case you have a chance to edit the comment in the setup dialog box.

3.2.2.12 Clear Screen - Clears the graphics screen and redraws the frame. Alternatively, click the "Clear screen first" button before hitting Plot when the selection has changed. This gives the appearance of any unselected traces disappearing while newly selected spectra appear and still-selected spectra remain unchanged.

3.2.2.13 Subtract from all – Subtracts the selected spectrum from all in memory (including itself). This is useful for subtracting a baseline, or for seeing the changes after an addition (to the cuvet) by subtracting the last trace before the addition. This operation subtracts from all spectra in memory, whether or not they are displayed on screen or listed (when there are more than spectra than can be listed). But the spectrum to subtract has to be in the list so you can select it..

3.2.2.14 Subtract from some – The last spectrum selected will be subtracted from all selected spectra. Unselected spectra are unchanged. This is for subtracting a baseline which applies to only certain of the spectra in memory. If you have already selected the set of spectra and the baseline was not the last, just deselect and select it again. The last selected will become a flat line at absorbance=0. More advice for this feature is given in the section “Making Figures”.

3.2.2.15 Save (binary, .mtx, or .csv). Saves the selected spectra to a series of binary spectrum files, a single .mtx ascii-format file, or a single .csv (comma-separated values) file. For the binary series, you will be asked for a basename and number for first in series. The spectra will be saved in the order they occur in memory. If you just want to save all spectra in memory, ***File: save all*** from the graphics screen is more convenient. For the .MTX file you can supply a filename and if desired select a subset (starting wavelength, ending wavelength, sampling interval) of the spectra to save.

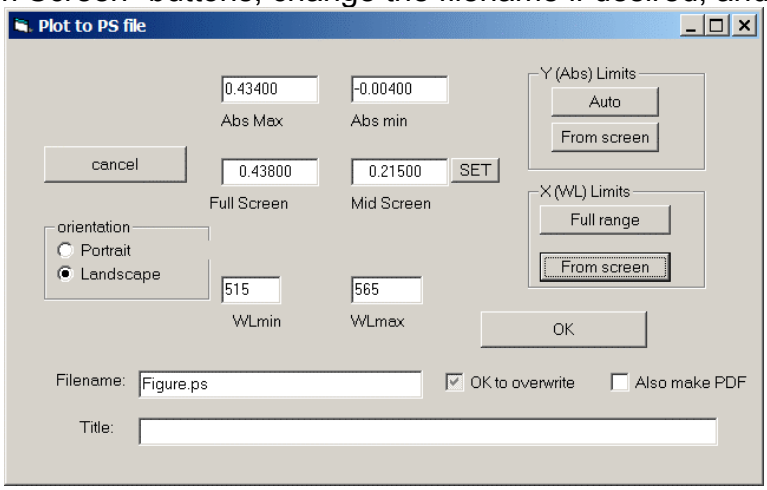
3.2.2.16 Multiply/divide - multiply or (if a box is checked) divide the selected spectra by a constant. This is useful if you want to scale a number of spectra by the same constant. For example you have a series of spectra from a given sample, which you know to have contained 2.3 µM of your compound, you can divide them all by 2.3 to get µM extinction spectra. Unlike the linear combination option, the spectra are multiplied in place rather than creating new spectra. The comment is modified to indicate the scale factor

3.2.2.17 Make Figure – Plots the **selected spectra** to a color postscript file. There is a great deal of selectivity for wavelength and absorbance limits, landscape vs postscript mode, and a possible descriptive title to print over the figure, but all is prefilled with sensible defaults. The postscript file is optionally converted to pdf if you have GhostScript on your computer (see Chapter 6

and appendix INSTALL/*SET-UP*). Otherwise it can be copied to a postscript printer, opened in PhotoShop to rasterize and make a .gif or tiff figure, or opened in a vector graphics program like Adobe Illustrator. If just a filename is given, the figure is saved in the current (data) directory. If an absolute path is given, it will be saved there. Reasonable defaults including a filename "Figure.ps" are preset, so you can just press *OK* to make some kind of figure. To reproduce the picture on screen, press *From Screen* buttons under X and Y limits. More details are given in Chapter 6, "Making Figures".

Make Figure

Select traces to include then click the "Make Figure" button to bring up the gui. If you already have the picture you want in the graphics screen, simply press both "From Screen" buttons, change the filename if desired, and press OK.



Four buttons on the upper right control the scale for the X (wavelength) and Y (absorbance) axes. Uppermost is "auto" which sets the absorbance scale to include all of the selected traces. However this is the default and the value has already been determined by the time this dialog comes up, so there is no need to click it unless you change the wavelength scale, in which case you need to click it to re-determine the absorbance scale for the new wavelength range. The second is "from screen" and this will change it to use the same absorbance scale that is currently used for the graphics display.

Below that is a button to set the wavelength range to full scale, but again that is the default so there is no need for this button except maybe to revert from an altered WL scale. Below that is a button to use the wavelength range currently in use on the display. I find it easiest to adjust the scale on the graphics screen to give the picture I want, then just click both "from screen" buttons to get the same in the figure.

In the upper middle are 6 boxes to display the current limits and for manually entering the absorbance and wavelength ranges. As on the "set limits" box for the graphics display, the second row is to set absorbance by midscale and fullscale values. If you modify the scale here you need to click "set" to transfer this to the max, min boxes above which are the final determinant of figure limits. In practice I hardly use these, because if I want to manually set limits it is easier to set them on the graphics screen (WYSIWYG) and use the "from screen" buttons to use the same values for the figure.

Filename, Title, portrait vs landscape (aspect ratio). Pretty self-explanatory. If you enter a Title it is printed just above the upper edge of the figure. These settings are retained after you make the figure or cancel (and the selected traces remain selected), so if the figure is not satisfactory you can easily come back, modify, and replot. However the scale limits are reset to default (fullscale wavelength and auto-fit for the selected traces) so if that is not what you want you need to modify these each time. And the settings that are retained will still be there if you make a completely new figure, so in particular be sure to change the filename.

3.2.2.18 Add a constant - Adds a constant value (given in the adjacent box) to all selected spectra, and moves the spectra in the display in real time. This is mainly for making figures, when it is the shape of the spectra that is important and the baseline irrelevant, to arrange the vertical position of each spectrum in the figure. Thus it is a more sophisticated version of the “stack spectra” option under display. For an example, we often monitor chromatography of cytochromes by taking three spectra of each fraction, one before and two after adding dithionite to see what is reduced immediately or slowly by dithionite. For a figure, we want to keep the three spectra of each fraction on the same baseline, but move the triplets for successive fractions to different levels. The three spectra of one fraction are selected and moved up or down together until there is no overlap with the previous triplet in the region of interest. For coarse movement put a relatively large number in the box (positive for up, negative for down), and keep clicking the button. The selected spectra move in real time. When the desired position is approached a smaller number can be entered to move up or down to the final place.

This works by plotting the spectrum in black to erase it, adding the constant, and plotting the result. As a result, when you move a spectrum up through other spectra in steps, each step will erase some pixels of the other spectra. To correct this, once the spectra are positioned, replot all. This does not affect the postscript plot, which is made from the final spectra in memory.

3.2.3 Drop-down menus on the Console (dashboard) window: "Other Functions"

3.2.3.1 *Resize.* Analogous to "resize" in the graphics screen display menu, this redraws the console window after it has been resized. There is no use in changing the width, but by increasing the height and clicking resize you get room for more spectra in the list. You may need to shorten it if working on a small screen and the buttons on the bottom are offscreen below.

3.2.3.2 superimpose spectra:
This is for comparing spectra that may have different scale, baseline offset or slanted baselines. It optimizes a slanted baseline and scale factor to optimally superimpose spectrum 1 on spectrum 2, with the ability to limit the fit to part of the spectral range.

3.2.3.3 Set up linear combination. When setting up a linear combination (under "operations" on the graphics screen), it can be tedious to select the spectra one by one. Select the spectra in the console, then select this menu item, The lin comb dialog will open with the selected spectra already entered, with all scale factors set to 1. If you want their sum, enter a comment if desired and just hit execute. Otherwise set the scale factors (positive or negative) for the desired combination and hit enter. To scale a single spectrum, without overwriting the original, select it, select this option, set the scale factor, offset, and comment, and click execute.

3.2.3.4 Fourier - currently does nothing

3.2.3.5 select all, deselect all - these are redundant with buttons at the bottom and are included here simply to assign shortcut keys (ctrl-A and ctrl-D) to these functions.

3.2.3.6 Set display limits - refers to the display on the graphics window, included here to define the shortcut ctrl-L so that this shortcut works regardless of which window has focus.

3.2.3.7 Three functions for *converting the Y scale* of spectra:
The selected spectra are unchanged, the new spectrum is put in a new trace. Log refers to common (base 10) logs.
Log A => A
A => log A
%T => A

Chapter 4. Data Formats and Import

4.1 Native formats

Two "native" formats are in use. Both allow saving a descriptive comment with each spectrum.

4.1.1. Binary files. The primary format for spectra is a binary format. Each file contains a single spectrum, but they can be loaded and saved in groups (called series). This is achieved by making the file extension a 3-digit number, the series number padded to three digits with leading zeros, like: myspectrum.001, myspectrum.002, etc (.000 is also allowed). (In the next major version (2.x) there will be changes to the binary format, and the filenames will be more conventional like myspectrum_001.spk or such. But we will still be able to read old formats)

4.1.2. Ascii Text files. These have extension .mtx (sometimes .mat, either is recognized). These are primarily used for grouping files that will be used together for analyzing spectra of mixtures (see Least-squares analysis). For example in analyzing pigments in plant extracts you might have in your spectral library a .mtx file with spectra of chlorophyll A, chlorophyll B, beta-carotene, and a couple of baseline fitters (See tutorial #2). However this format can also be useful in other contexts. It is also more flexible in handling sampling intervals that are not submultiples of 10 (1, 2, 5, 10 points per nm). A more complete description of the formats is provided in appendix 2. Instructions for associating these extensions with the scanedit program are in "Customizing scanedit". Appendix "installing and setup".

Both formats can be loaded from within scanedit using the File:open menu (3.1.3.1.1), by dragging the file and dropping into the "drop files here" pane on the graphics window (3.1.2.1.), by dragging the file onto the icon (or a shortcut) of scanedit (to start a new instance and load the files), or, if you have set up file associations, by double-clicking on the spectrum file.

ScanEdit requires that all spectra in memory at one time have the same sampling interval, starting wavelength, and number of points. This is because most of the operations that scanedit carries out involve arithmetic manipulation of the spectra, and this is only possible when they are sampled at the same points. If you think of each digital spectrum as a vector, i.e. "ordered multiplet", then the operations are vector operations, and only make sense if all the spectra exist in the same vector space. Some arrangement for loading spectra with different wavelength range is made in the "Load Into" operation on the file load dialog (3.1.3.1.1). Otherwise, if you try to load a spectrum with different sampling from what is presently in memory, you are given the option to abort the operation or delete current spectra and continue. Of course you can modify the wavelength range and sampling intervals of each spectrum to include only the points present in all the spectra, and then load them simultaneously.

By the way, the program is not tied to any particular wavelength unit. We mention "nm" throughout this text because that is the standard unit for UV-vis spectra, but it could be anything. We even have cases where it was degrees of two-theta (for analyzing x-ray diffraction powder patterns) or seconds (for analyzing reaction time courses.

4.2 Importing Other Formats

Ideally you should be able to effortlessly process the data from your spectrophotometer by double-clicking, dragging and dropping or opening the files from a menu. In fact the programs in this suite require a particular binary format of data ("native format"). Import utilities are available for the .spc files of the Shimadzu UVPC program and for Olis XTML files, as well as plain-text and comma-delimited formats described below. If your spectrophotometer uses another common format, the developer would be glad to work with you to prepare import utilities for your format- I would need files of several spectra together with a description and or plots of the spectra. Until then, if your spectrophotometer does not save spectra in a file we can import, you will need to export the spectra to some ascii format (plain text or spreadsheet) and prepare one of the following:

4.2.1 Importing Shimadzu UV-PC .spc files.

Note this is a different format from the .spc files saved by the newer Shimadzu UV-Probe software, or from the Galactic .spc format. The .spc files are almost like a native format for scanedit, the main distinction being that they are loaded from the file:import instead of file:open menus. They can also be opened by dragging onto the "drop files here" pane or onto the icon of scanedit. Like the native binary format, they contain a single spectrum but can be grouped in a series as follows. The UV-PC program saves each spectrum, together with a descriptive comment, in a file with a filename of up to 8 characters, and the extension .SPC. If you make 8-character filenames with the first 6 being all the same and the last two being consecutive 2-digit numbers, scanedit will treat it as a series and load all when the first is selected to import. It is convenient to generate a list of filenames in notepad. As each spectrum is taken, a comment is entered after the filename and the whole line is copied and pasted into the filename box (which only take the first 8 char) and the comment box (so the original filename will be included in the comment).

Importing data from spreadsheet or JCAMP-DX files.

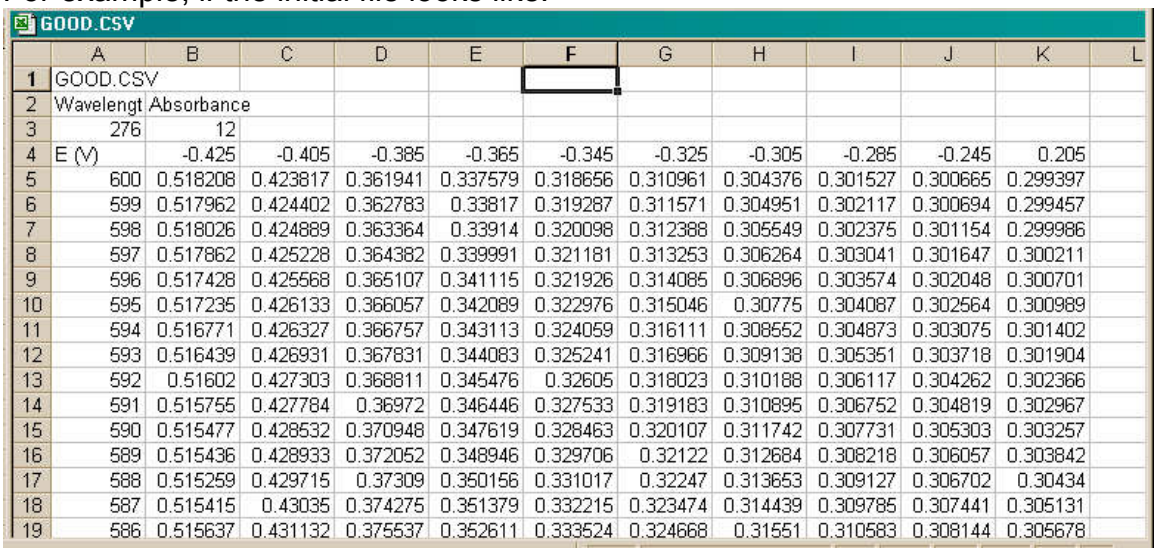
4.2.2. CSV (comma-separated values) files. (3.1.3.1.5, "import csv" option in scanedit):

"Spreadsheet files (.xls(x)The file can contain one or more spectra, all sampled at the same wavelength values. The spectra should be stored as columns in the spreadsheet or text file, with wavelength values in the first column. Wavelength can be increasing or decreasing as you go down the column, but the sampling interval should be constant. (If the sampling interval is not constant, you can easily convert it to JCAMP format and load that with resampling- see next section 5 on JCAMP files.)

4.2.2.1. Preparing the data.

Open the file in Excel. Replace any header rows above the data with just one row, consisting of "Wavelength" in the first field (above the column of wavelength values) and above each of the remaining columns, a comment describing the spectrum in that column. Even if you don't care about the comments, it is important to have at least something in the field above the last column, and nothing after that in the row. This way when the file is saved in comma-separated-values format, the first line will have a number of commas equal to the number of columns minus 1, which is taken as the number of spectra. (If things

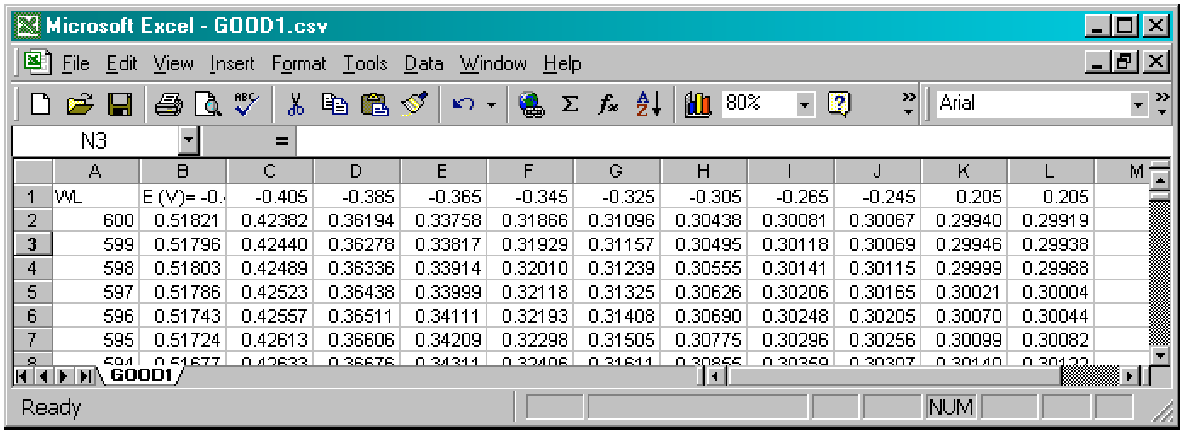
don't "just work" you can verify/fix this by opening the csv in a text editor and counting commas.)
The comment can be up to 64 characters long and can contain spaces but no commas (for input from a .csv)(Commas may be OK if entire comment is enclosed in quotes). In general the comment can be blank, but in a csv file better put something, at least for the last spectrum.
For example, if the initial file looks like:



GOOD.CSV

	A	B	C	D	E	F	G	H	I	J	K	L
1	GOOD.CSV											
2	Wavelengt	Absorbance										
3	276	12										
4	E (V)	-0.425	-0.405	-0.385	-0.365	-0.345	-0.325	-0.305	-0.285	-0.245	0.205	
5	600	0.518208	0.423817	0.361941	0.337579	0.318656	0.310961	0.304376	0.301527	0.300665	0.299397	
6	599	0.517962	0.424402	0.362783	0.33817	0.319287	0.311571	0.304951	0.302117	0.300694	0.299457	
7	598	0.518026	0.424889	0.363364	0.33914	0.320098	0.312388	0.305549	0.302375	0.301154	0.299986	
8	597	0.517862	0.425228	0.364382	0.339991	0.321181	0.313253	0.306264	0.303041	0.301647	0.300211	
9	596	0.517428	0.425568	0.365107	0.341115	0.321926	0.314085	0.306896	0.303574	0.302048	0.300701	
10	595	0.517235	0.426133	0.366057	0.342089	0.322976	0.315046	0.30775	0.304087	0.302564	0.300989	
11	594	0.516771	0.426327	0.366757	0.343113	0.324059	0.316111	0.308552	0.304873	0.303075	0.301402	
12	593	0.516439	0.426931	0.367831	0.344083	0.325241	0.316966	0.309138	0.305351	0.303718	0.301904	
13	592	0.51602	0.427303	0.368811	0.345476	0.32605	0.318023	0.310188	0.306117	0.304262	0.302366	
14	591	0.515755	0.427784	0.36972	0.346446	0.327533	0.319183	0.310895	0.306752	0.304819	0.302967	
15	590	0.515477	0.428532	0.370948	0.347619	0.328463	0.320107	0.311742	0.307731	0.305303	0.303257	
16	589	0.515436	0.428933	0.372052	0.348946	0.329706	0.32122	0.312684	0.308218	0.306057	0.303842	
17	588	0.515259	0.429715	0.37309	0.350156	0.331017	0.32247	0.313653	0.309127	0.306702	0.30434	
18	587	0.515415	0.43035	0.374275	0.351379	0.332215	0.323474	0.314439	0.309785	0.307441	0.305131	
19	586	0.515637	0.431132	0.375537	0.352611	0.333524	0.324668	0.31551	0.310583	0.308144	0.305678	

You could edit it like:



Microsoft Excel - GOOD1.csv

File Edit View Insert Format Tools Data Window Help

N3

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	WL	E (V)= -0.	-0.405	-0.385	-0.365	-0.345	-0.325	-0.305	-0.265	-0.245	0.205	0.205	
2	600	0.51821	0.42382	0.36194	0.33758	0.31866	0.31096	0.30438	0.30081	0.30067	0.29940	0.29919	
3	599	0.51796	0.42440	0.36278	0.33817	0.31929	0.31157	0.30495	0.30118	0.30069	0.29946	0.29938	
4	598	0.51803	0.42489	0.36336	0.33914	0.32010	0.31239	0.30555	0.30141	0.30115	0.29999	0.29988	
5	597	0.51786	0.42523	0.36438	0.33999	0.32118	0.31325	0.30626	0.30206	0.30165	0.30021	0.30004	
6	596	0.51743	0.42557	0.36511	0.34111	0.32193	0.31408	0.30690	0.30248	0.30205	0.30070	0.30044	
7	595	0.51724	0.42613	0.36606	0.34209	0.32298	0.31505	0.30775	0.30296	0.30256	0.30099	0.30082	

GOOD1

Ready NUM

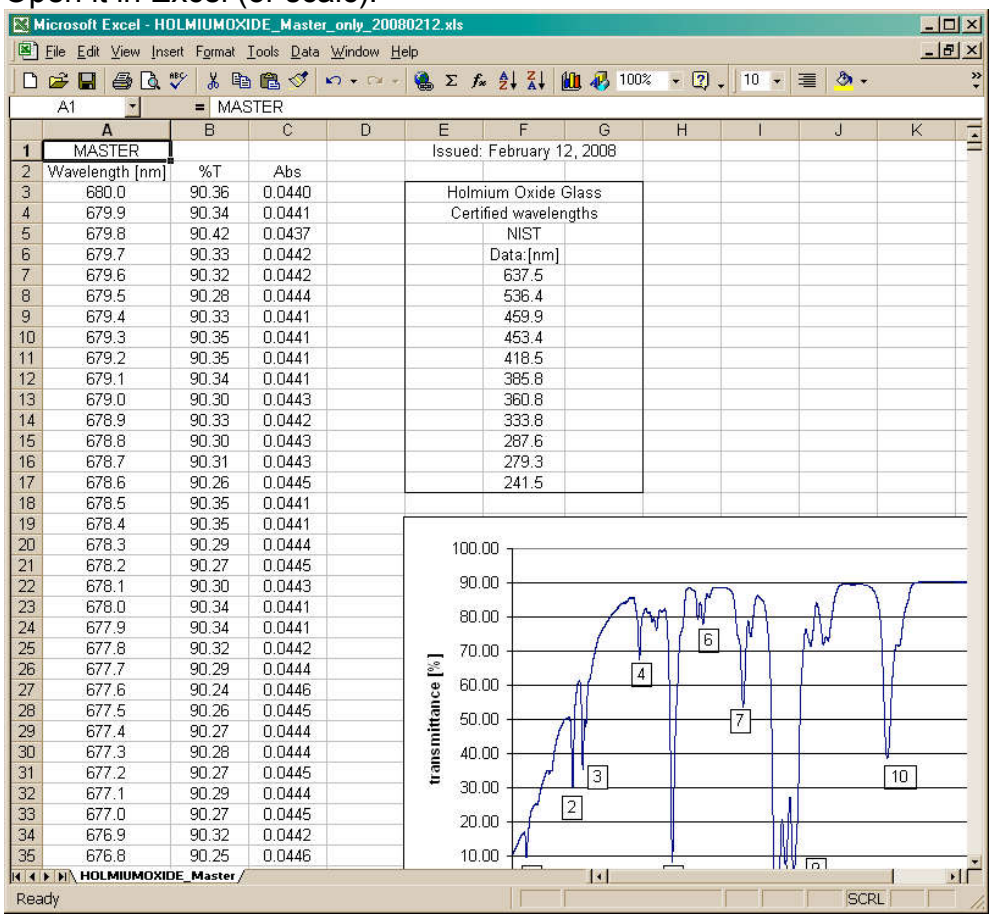
*****Be sure the cells are formatted to display all significant figures- if they are rounded off in the spreadsheet display, they may be rounded off when you save the .csv file. It is ok if the csv contains scientific notation (1.2345E-01).

4.2.2.2. loading the data:

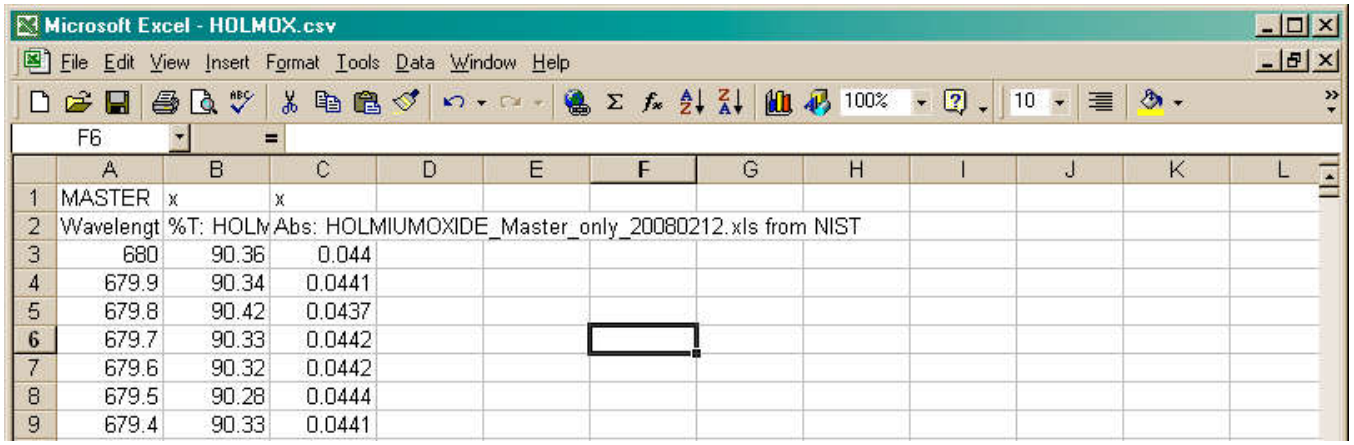
Save as .csv and open from within the program using File:import:import csv (first select the file and then click "import csv". Or drag its icon onto the scanedit icon to start the program and load the data. There will be a message "The file contains x spectra . . . This will give a clue whether the spectra will load correctly, and what is wrong if they do not. Examine the spectra to see if conversion succeeded, and save in the native binary or text (.mtx) format. If not, and especially if the wrong number of spectra were announced during conversion, open the .csv file in a text editor to check.

4.2.2.3. Other examples:

4.2.2.3.1. Get the spectrum of Holmium oxide glass in excel format from:
https://www.nist.gov/document/holmiumoxidemasteronly20080212xls
Open it in Excel (or scalc):

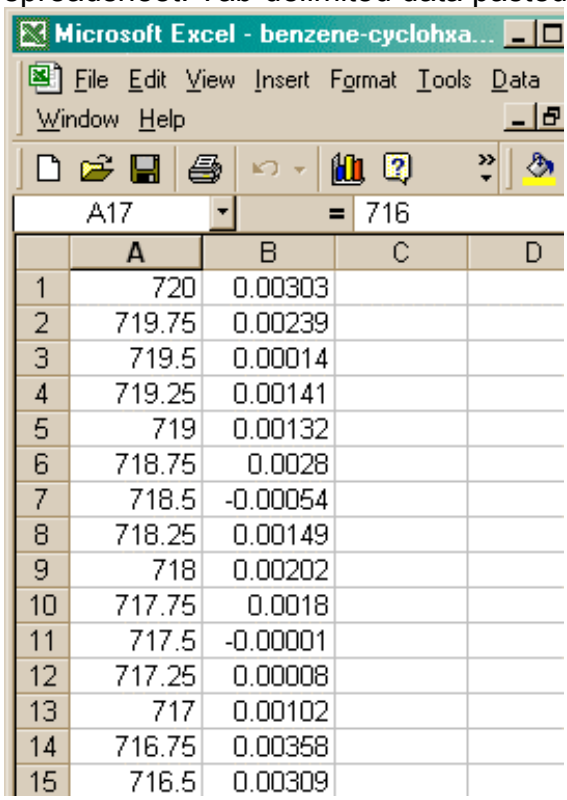


First delete the charts and everything but the first 3 columns
The first column is wavelength, second is percent T, and third is absorbance.
We could just delete the second column since scanedit is geared toward absorbance or at least values that satisfy something analogous to Beer's law. But in case it is useful, leave it in.
There are two rows above the data. Delete the first ("Master") and keep the second as comments: %T and Absorbance.



The fields of the Header row will be taken as “comments” or descriptive labels, for the spectra. The comment can be up to 64 characters and in general can contain any punctuation, but commas will probably make a problem for a csv file. (Putting quotes around the label that contains commas might work, feel free to experiment.) You can leave %T and Abs or put something more detailed to help you remember what it is, where it came from.

4.2.2.3.2. A third example- get a spectrum from the photochemcad package (http://www.photochemcad.com/). Original data and figures are available at: http://omlc.org/spectra/PhotochemCAD/html/index.html. (http://omlc.org/spectra/PhotochemCAD/abs_txt/benzene.abs.txt)*. Select the entry "benzene in cyclohexane". Scroll down below the figures and click the link "original data" – Save the page (or the link target) as a textfile. This is a tab-delimited text file. You should be able to get excel to open it as such, but if it doesn't work (if it opens as a single column with both wavelength and absorbance in one field), open it in word, select all, and paste into a blank excel spreadsheet. Tab-delimited data pasted into excel goes into separate fields.



	A	B	C	D
1	720	0.00303		
2	719.75	0.00239		
3	719.5	0.00014		
4	719.25	0.00141		
5	719	0.00132		
6	718.75	0.0028		
7	718.5	-0.00054		
8	718.25	0.00149		
9	718	0.00202		
10	717.75	0.0018		
11	717.5	-0.00001		
12	717.25	0.00008		
13	717	0.00102		
14	716.75	0.00358		
15	716.5	0.00309		

The file starts with data in the very first line- no header. Insert one blank line above the raw data, put WL (or

	A	B	C	D
1	WL	Benzene in cyclohexane		
2	720	0.00303		
3	719.75	0.00239		
4	719.5	0.00014		
5	719.25	0.00141		
6	719	0.00132		
7	718.75	0.0028		
8	718.5	0.00054		

anything, it will be ignored) in the first field, and a descriptive comment (Benzene in cyclohexane) in the second field. Save it as a csv-formatted file with extension .csv.
Select and load the file in scanedit with **“Files : import : import csv”**. Save in .mtx form. The points are sparse, so these spectra look best with linemode set to lines (the default), not points.

Troubleshooting: Everything should be fine, but if not examine the .csv file with a text editor like notepad (MSWord will do). scroll down and look for irregularities. If part of the file is read correctly but it terminates early, check that none of the lines end with ",". There should be commas between the numbers but not after the last one. If there are, delete them and save the file (text format) and with extension .csv (not .csv.txt or .csv.doc). Early in the import process, the number of spectra found in the file is reported. If this is wrong, check that the number of commas in the first line is equal to the number of spectra (n+1 fields separated by n commas).

4.2.2.3.3 Tabular text data -
In MS Word you can replace "white space" with tabs. Select the table, paste it into excel and the tab-separated numbers will go into successive columns. Then proceed as above to fix the header we need.

4.2.2.4. Notes

Any spectra already in memory before importing .csv are discarded (A warning is issued first). At the start the number of WL points is unknown and it is highly unlikely to be the same as what is in memory now, so memory is cleared before starting. If you have spectra in memory you can first save them, or open a second instance of the editor to load the .csv file and save the spectra in native format. Because the spectra are stored as columns in the .csv, it is not possible to process one spectrum at a time with a single pass; the whole block must be read into memory to effectively transpose the data into rows. Thus there is no option to convert without loading as for the Shimadzu or integer binary files.

Wavelength sampling interval: The csv import does not "re-sample" the data, so wavelength intervals must meet the requirements of scanedit.

This has largely been automated now, so first try just importing- for example the benzene spectrum in the last example automatically changes ppnm to accommodate 4 points/nm. But if there are problems the following might be helpful.

The sampling interval must be constant within a spectrum, and only spectra with the same sampling interval can be loaded at any one time. For historical reasons, in the current version the sampling interval is defined by two integers: a fundamental highest sampling rate (by default 10 points /nm) , and the actual sampling rate is expressed as a multiple of the resulting interval (by default 0.1, 0.2, 0.5 or 1 nm. When you load a spectrum in which the sampling rate is not an even multiple of 0.1 nm, that fundamental sampling rate is changed, for example to 8 points per nm if the sampling interval is 0.25 or 0.125 nm or 20 if the interval is 0.05 nm.

This all happens automatically and shouldn't concern you, except for one thing: The current binary format does not save the fundamental frequency, because at the time that was invariant. So if you import a file with 0.25 nm spacing, the frequency switches to 8 points/nm (ppnm), and everything is fine. But if you save this as a binary file, restart scanedit and load the file, the sampling interval will be seen as 0.2 nm and the wavelength scale will be off. You can fix this if you know what the sampling interval should be by going to the file menu on the graphics screen and selecting "redimension". Change the item labeled ppnm-max (to 8 in this case) and OK. This does not redimension the array or erase your spectra from memory, it only affects the wavelength scale.

But it is better to save anything with a non-standard sampling interval in the ascii .mtx format, which records the actual sampling interval (appendix 2), and when loaded, ppnm is set to display the wavelength correctly, changing ppnm if necessary.

Fixing this problem is straightforward, but it requires a change in the format of binary files and resulting incompatibility with earlier files, so it and other improvements are being postponed until the next major version (2.0) which should be out soon.

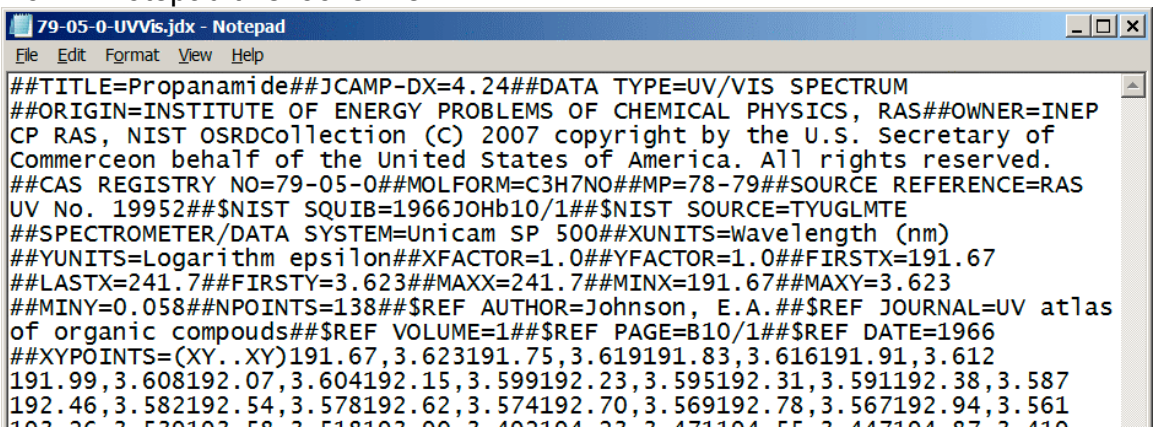
4.2.3. JCAMP spectra, and other spectra with variable sampling interval.

In olden times, when disk space was more precious, or when digitization of spectra required human effort proportional to the number of points, it was common to sample spectra finely in areas where the absorbance was changing rapidly, as around spectral peaks, and sparsely elsewhere. Now a routine has been provided to load these spectra into temporary space and resample, i.e. interpolate values at even sampling intervals. Although it is designed for JCAMP format, it only requires two tags of that format, so it is easy to convert a spreadsheet file to a format that will be recognized as JCAMP and resampled. The data in JCAMP files is often stored as "Log Epsilon", while scanedit procedures are geared toward working with absorbance, so a procedure for changing log epsilon to epsilon is provided also(3.2.3.7).
JCAMP support is fairly new and not guaranteed to work with all JCAMP spectra. The file should contain only a single spectrum.

4.2.3.1 Background
The JCAMP-DX format is as follows:
a. A header consisting of tags indicated by starting with ##. The only tags we are using here are ##TITLE=, ##YUNITS=, ##XYPOINTS=, and ##END. Anything else (besides the data) is ignored. The first two are optional but will be combined to make the comment if present. The ##XYPOINTS is required since it tells where the data starts (right after this tag).
b. This is followed by wavelength, absorbance pairs, with a comma between.
c. After the last point, ##END= signifies end of the data.
Many JCAMP files have unix-type newline characters (line feed; \x0A) while scanedit needs MSWindows-style newlines (CRLF; \x0D\x0A). If you download the JCAMP file and look at it with notepad, it is all one big block of text with no carriage returns. There are several ways to convert, but on windows the easiest is to
1. load the .jdx into wordpad or MSWord and save as "plain text", with a different name if you like.
2. Load the .jdx file into MS Excel (from file:open in excel, not drag - drop). In the import dialog that comes up, specify type: delimited, then delimiter: comma
This should give a number of header lines in the top rows, then two columns: wavelength and absorbance (or log absorbance, or %T or some such spectral measurement - this should be specified in the header). Make sure there are actually two columns, and not both values in the first column and the second column blank). Save in .csv form and import as JCAMP.
3. If you have a spectrum with variable spacing that is not a .jdx file, you cannot load it with import .csv" because that assumes constant sampling interval. But you can easily convert it into a jdx file and load it as such. The data should be in two columns- first wavelength, then absorbance value. Just put two lines above the data: ##TITLE= with an optional descriptive comment; and ##XYPOINTS= immediately above the data. Then put ##END= after the last line of the data.

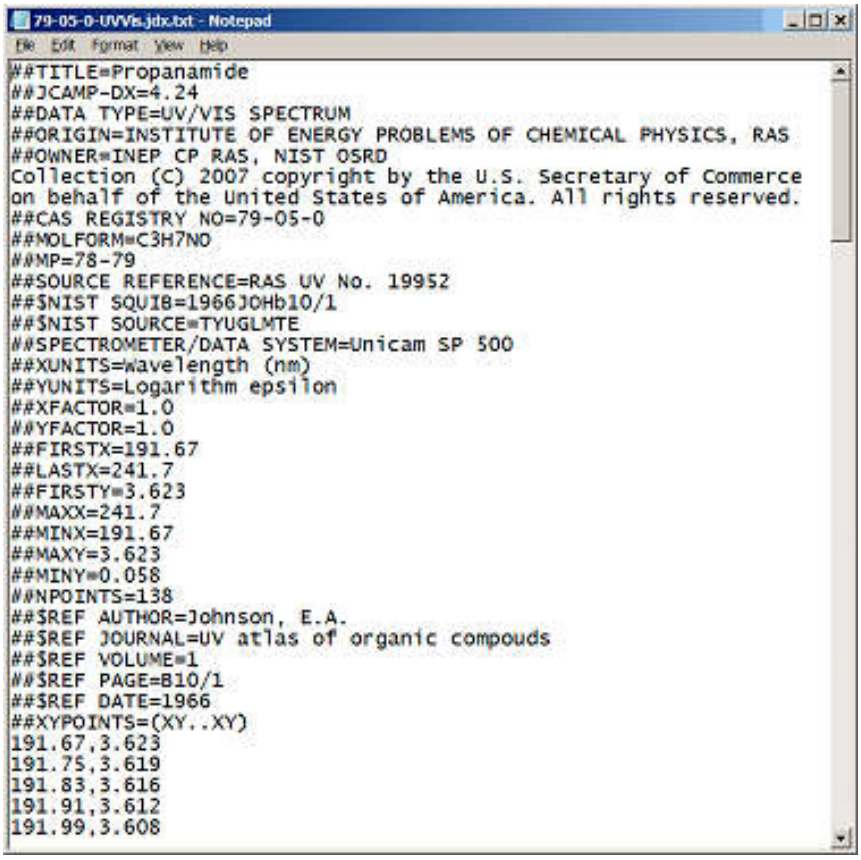
4.2.3.2 Example 1: propionamide from
https://webbook.nist.gov/cgi/inchi?ID=C79050&Mask=400
scroll down below the UV spectrum and click "Download spectrum in JCAMP-DX format. " (or:
https://webbook.nist.gov/cgi/inchi?JCAMP=C79050&Index=0&Type=UVVis)

Now in notepad this looks like:



Load in wordpad to convert newlines, save as plain text in .jdx.txt:

Examine in notepad to verify CR's have been inserted:



In scanedit, from the dropdown File menu, click "import" to open the import dialog. Navigate to where the jdx.txt file is and select it, then click JCAMP.

The spectral data are read in, and a new spectrum is generated by interpolating between the points. to constant-interval wavelength values. Only linear interpolation is done now, so if the original sampling is too sparse you may notice a polygonal-outline spectrum: straight lines between the original points. But if the source spectrum had enough points to define the shape of the spectrum, this should be a good approximation. In this case the spectrum is sampled quite finely, at nearly constant intervals.

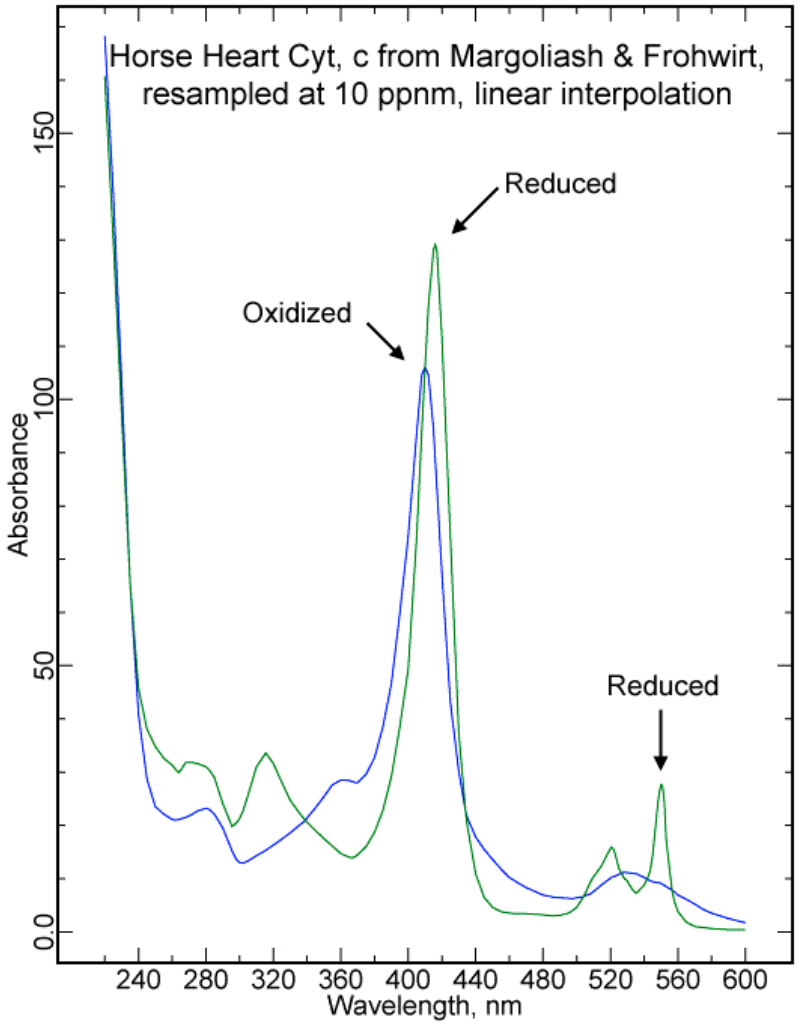
As you see from the original header, the units of the Y-values are "Log epsilon", so convert to Epsilon (Absorbance of a solution at unit concentration) before doing any spectral algebra. Open the console and select the spectrum by checking its box. From the menu "Other Functions" at the top of the console window, select Log S -> A. You are prompted for a number to subtract before exponentiation, like 0 for Molar epsilon, 3 for mM, 6 for µM. Enter a number (or leave blank, equivalent to entering zero) and hit OK. The spectrum is calculated in the next available trace.

Hint- if the resulting spectrum contains very large values, because it was log epsilon for a strongly absorbing compound and you didn't subtract 3 or 6, but the scale of the graphics screen is set for fullsale 2A, the program may crash with overflow calculating the pixel position of these points. To avoid this, before converting set the absorbance scale to from -1E6 to +1E6. With that there is no problem with very high absorbance values, except that the absorbance labels on the left may not fit in their space.

4.2.3.3 Example 2: formatting an arbitrary spectrum to be read as JCAMP-DX: Get the spectra of Horse Heart cytochrome c in oxidized and reduced form from an article by Margoliash and Frohwirt: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1196833/pdf/biochemj00823-0163.pdf The wavelength points are not spaced at constant intervals, but selected to best define the spectrum with a small number of points: close together at peaks, widely spaced between.

OCR or enter by hand the values from the table. Here we have tab-separated text with both oxidized and reduced spectra

Wavelength	Cytochrome c	
(nm)	Reduced	Oxidized
220	160.7	168.3
225	130.3	138.8
230	97.3	103.1
235	65.7	65.7
240	45.7	40.8
245	38.1	28.7
250	34.8	23.5
255	32.6	22.1
260	31.3	21.1
262	30.5	21.0
264	29.9	21.1
268	31.8	21.5
270	31.8	21.8
272	31.8	22.1
275	31.6	22.7
280	31.0	23.2
282	30.4	23.0
285	29.0	22.1
290	24.1	19.5
295.5	19.8	15.2
298	20.4	13.6
300	21.3	13.0
302	22.7	12.9
305	25.7	13.3
310	31.0	14.3
315.5	33.6	15.3
320	31.6	16.3
325	28.1	17.4
330	24.7	18.6



335	22.5	19.9
339	20.9	20.9
340	20.6	21.4
345	19.0	23.3
350	17.5	25.4
355	16.1	27.6
360	14.7	28.5
362	14.4	28.5
365	14.0	28.4
367	13.9	28.2
369	14.2	28.0
370	14.5	28.0
375	16.0	29.6
380	18.7	32.7
385	22.8	38.5
390	29.1	46.6
395	38.5	59.8
400	49.3	74.3
405	74.1	93.6
408	93.7	104.6
410	106.1	106.1
412	117.6	104.6
415	128.1	94.4
416	129.1	88.8
417	128.0	83.5
420	111.6	67.3
425	74.2	43.1
430	37.3	30.1
434	22.7	22.7
435	20.0	21.5
440	10.9	17.8
445	6.5	15.5
450	4.6	13.6
455	3.8	11.8
460	3.5	10.2
465	3.4	9.2
470	3.4	8.3
475	3.3	7.6
480	3.2	6.9
485	3.0	6.6
486	3.0	6.5
490	3.1	6.4
495	3.5	6.3
497	3.9	6.2
500	4.6	6.4
504	6.6	6.6
505	7.2	6.8
508	9.3	7.1
510	10.4	7.6
512	11.2	8.1

515	12.6	8.9
518	14.5	9.6
520.5	15.9	10.2
522	15.4	10.4
523	14.3	10.5
525	11.9	10.8
526.5	11.0	11.0
528	10.1	11.2
530	9.5	11.1
533	7.9	11.0
535.25	7.2	10.9
540	8.8	10.2
541.75	9.9	9.9
543.5	11.4	9.6
545	14.3	9.4
547	20.6	9.3
548	23.9	9.2
549	26.3	9.2
550	27.6	9.1
550.25	27.7	9.0
551	27.1	8.9
552	24.2	8.7
553	17.5	8.5
555	11.8	8.1
556.5	7.8	7.8
558	5.6	7.5
560	3.8	6.9
565	1.8	6.1
570	1.0	5.2
575	0.8	4.2
580	0.6	3.5
590	0.4	2.5
600	0.4	1.7

The sampling interval is not constant, so we can't load this as a .csv file. To load as JCAMP, you need to delete one column: the current implementation can only read one spectrum at a time in this format, so prepare two files, one with the oxidized spectrum and one with the reduced.
So load this in Excel as a tab-separated file. delete one column (2nd or third). Delete everything above the values and enter two keys before the data, and ##END= after, and save as csv (comma-separated values).:

```
##TITLE E. Margoliash, N. Frohwirt and E. Wiener 1959 Reduced cyt c
##XYPOINTS 111
220, 160.7
225, 130.3
230,  97.3
235, 65.7
240, 45.7
. . .
```

580, 0.6
590, 0.4
600, 0.4
##END=

Start scanedit, File:import: (navigate to the csv file and select it): click JCAMP, and you have the spectrum Since this was in absorbance already, no need for "logE => e". There is some polygonization of the peaks due to sparse sampling of the original data, but the overall shape is correct:

GETTING YOUR DATA INTO THE RIGHT FORMAT-

Ideally the programs should allow you to effortlessly process the data from your spectrophotometer by dragging and dropping or opening the files from a menu. In fact the programs in this suite require a particular binary format of data ("native format"). Import utilities are available for the .spc files of the Shimadzu UVPC program and for Olis XTML files, as well as plain-text and comma-delimited formats described below. If your spectrophotometer uses another common format, the developers would be glad to work with you to prepare import utilities for your format- we would need files of several spectra together with a description and or plots of the spectra.

Until then, if your spectrophotometer does not save spectra in a file we can import, you will need to export the spectra to some ascii format (plain text or spreadsheet) and prepare one of the following:

Chapter 5. Customizing ScanEdit

5.1. Getting ScanEdit - Download the file and if necessary register the comdlg32.ocx file as described in appendix A; and verify that it runs on your computer.

5.2. Put the program in its place:

You can run the program from any location you have access to, but if multiple users will use it, it would be good to put in an accessible place. Make a folder "scanedit" in the "C:\Program files" folder (or C:\Program files(86) with a 64-bit operating system) and copy the executable there. Actually you can put it wherever you like, so you don't need administrative privileges to set it up unless you have to install the .ocx file as mentioned above.

5.3. Make a shortcut on your desktop:

ScanEdit can be started by dragging a file and dropping it on the icon of scanedit or its shortcut. If the file is one that scanedit can open, it will be loaded, and if it is part of a series the subsequent members will be loaded. But even if it is not a file scanedit can use, the working data directory will be changed to the directory of that file.

To make the shortcut, open the directory with scanedit and drag the scanedit icon and, while holding down the <Alt> key, release it on your desktop. Or, right-click on the desktop and select "new"; then "shortcut". Navigate to the location of the executable and select it.

5.4. Program associations:

You can associate certain filetypes with the programs so that double-clicking the file's icon (or typing its name in a shell) opens the file with the chosen program. Windows associates filename extensions with "ftype" file types, and associates ftypes with the program that opens them. We define our two formats with separate ftypes, but since they all open with the same program you could just make one. Here is our scheme:

extensions	ftype	program
.000, .001	binspec	scanedit
.mat, .mtx	ascispec	scanedit

Unfortunately .mat is somewhat overused, already by MS access and MatLab, and if the .mat extension *is* associated with MS Access, the extension will not be shown on filenames even if you have unchecked "hide extensions for know filetypes" in Folder Options. Therefore if you use Access .mat files, it would be better to assign .mtx for ascispec files. Be aware however that the documentation and the program may at times mention .mat files. In the next version we will transition completely to .mtx.

```
     ftype binspec=D:\path\to\Scanedit.exe %1
     ftype ascispec=D:\path\to\Scanedit.exe %1
rem  assoc .mat=ascispec
     assoc .mtx=ascispec
     assoc .000=binspec
     assoc .001=binspec
```

Since the binary spectra will generally be loaded as a series, it is only necessary to associate the first in the series (.000 or .001). If you put these in a batch file, double the “%” characters so they are not stripped by the shell. Otherwise just copy and paste at the command line.

5.5. Environment preference settings:
You can customize certain things like where the program initially looks for data or library spectra, Font for the plots, colors for plots and onscreen, etc. Currently this is done by setting environment variables. In the next version these can be set under the preferences menu and saved in your "appdata" directory to be read in each time the program starts, but for now most configurable preferences are set by setting environment variable.

SE_datadir (full path including drive)
Default location to look for spectra if started by double clicking or typing the name of the executable (If started by dragging a spectrum file onto the shortcut or double-clicking a spectrum file, it will start in the folder with that file).
SE_libdir
Your spectrum library: Location to look for .mat or .mtx file with standards to use in fitting experimental spectra.
SE_GSexec (path\filename for gswin32c.exe (ghostscript for pdf)
SE_PSfont (Font to use in postscript figures.) If you want to use your windows fonts in the figures instead of generic equivalents, also set GS_FONTPATH=%WINDIR%\FONTS\ . See the documentation on making figures for more information.
Colors for the PostScript plots and for plotting on screen are now saved in a file in appdata, and can be edited using options under the "preferences" menu. See Preferences: edit plot colors (3.1.3.8.4-5)
SE_LineMode=[DOTS|LINES|AUTO]

You can set variables for your account or (with privilege) for the machine by right-clicking “computer” ("This PC"), selecting properties, then clicking the “advanced” button (advanced system settings on Win7-11), then the environmental variables button.
If your account does not have administrative privileges, on windows vista or above you need to edit environment variables through the “User Accounts” control panel: User Account : user accounts : change my environment variables.
You can also set env variables with the "set" or "setx" commands. To see what has been set, type set with the first two letters SE:

```
C:\>set SE
SE_datadir=G:\writing\spectral\Taylors
SE_GSexec=C:\Program Files (x86)\gs\gs9.10\bin\gswin32c.exe
SE_libdir=g:\standard
SE_LineMode=DOTS
SE_PSfont=Arial
```

5.6. Setting linemode - (3.1.3.8.1) This affect how spectra are drawn. When the sampling interval is less than the width of a pixel, and the curves are not very

steep, just setting one pixel for each data point works well. But with sparse sampling, or when zoomed in, or when the curve is very steep, this results in dotted lines, and in that case it is better to draw lines between the points. The default is to draw lines. In principle drawing dots is faster, but in fact the difference is not noticeable.

With "set linemode" under the "preferences" menu on the graphics window there are three options- always draw dots, always draw lines, or "auto". "auto" draws dots when the sampling interval is less than the pixel width (without considering steepness of the curve) and draws lines otherwise. The default is to always draw lines. (correction- "Auto" means draw lines if the number of points plotted across the graphics screen is less than 500. If the graphics screen is about 500 pixels, the result is the same)

The screen is not automatically re-drawn after making the change, so you won't see any difference until a spectrum is drawn. The changes made here only affect the current session. To set the default, set the env variable SE_LineMode=[DOTS|LINES|AUTO]

5.7. Setting up the GhostScript program (GS) to convert .PS to .pdf files. Download GS from: https://ghostscript.com/releases/gsdnld.html . ScanEdit has been tested with versions 9.09 and 10.07, 32 or 64 bits. Get the free AGPL-licensed version unless you are using this commercially. Install it according to the instructions. Notice what folder the program is installed in (or “get properties” on the shortcut that is installed in the start menu or on the desktop).

Then run scanedit and select “locate GSWin32c.exe” under the Preferences pull-down menu (3.1.3.8.2). This opens a file-selection dialog with which you can navigate to the folder where GS is installed, and locate GSWin64c.exe: typically C:\Program Files\gs\gs9.10\bin\gswin64c.exe Select the file. This will be used for the current session and (on Win7 or later) will be saved to the user's environment and loaded at startup for future instances. To make this permanent on older versions, set the environment variable SE_GSexec to the full path and filename, as described above (5.5).

In the next release all preferences will be set from inside scanedit and saved in a file in the users application data folder, rather than by environment variables. As a start on that, the color pallets for drawing spectra are already saved there, if you edit the pallet using the color picker (preferences) and choose to make the changes permanent

5.8 Complications introduced by dynamic dimensioning :

Although memory is seldom limiting now, ScanEdit maximizes the use of memory by dynamically dimensioning the data array. when a new spectrum is loaded, the array is redimensioned for spectra of that many points and as many spectra of that length as will fit in the total number of 250,000 data points. Thus, if your spectra run from 250 to 650 nm with 10 points per nm (4001 points), you can only load 62 spectra.

However this number 250,000 is arbitrary. I don't want to set it too high for fear that on some machines the program will crash stillborn when it tries to dimension the array. Instead I put an option under preferences to change this value. On a Win7 machine with 8GB of memory, I set the max value to 2.5 million

and load a spectrum from 100 to 1000 nm (9001 points, and it dimensioned for 200 spectra of that size (there is a limit of 200 on the number of spectra that can be loaded).

Still that is 1.8 million data points. So if you really need more spectra, use this option. In the future it will be one of the "preferences" that can be saved in %appdata%. If you are testing this, note the array is not redimensioned as soon as you change this number, but only when redimensioning is triggered by loading spectra of a different length, or by selecting redimension under the File menu and changing pmax. Also note that any spectra in memory are erased on redimensioning.

The redimensioning works transparently for almost everything. Other operations force the requirement that all spectra in memory have the same wavelength range and sampling interval. Therefore if you have spectra of a certain length in memory, and the array has been redimensioned to just accommodate that length, if you then want to load longer spectra you will have to abandon the short spectra, and the array will automatically be redimensioned for the longer spectra.

However there are two functions which allow to partially get around that requirement, and here the dynamic dimensioning introduces a complication. The "load into" (3.1.3.1.1) function allows you to modify the spectrum being loaded to conform with the current spectral range. Portions of the spectrum outside the current spectral range (or a smaller subset that you select) are truncated. If the new spectrum or the subset of it that you have selected does not cover the current spectral range, the missing regions are filled in with zero absorbance if you are loading into a new trace, or with the values of the current spectrum if you are loading into it. This works fine except when the array holding the current spectra is dimensioned for a number of points too small to hold the spectrum being loaded in its original form you will get a message that the array dimension is too small to load the new spectrum, and the load aborts. (This problem could be eliminated in a future release by first loading the spectrum into another array that could be separately dimensioned). For now the work-around is to first load the longest spectrum that will be used. This sets the dimension to accommodate it. Then, to inhibit redimension when you load the shorter spectra, and select "redimension" under the file menu. Check the box labeled "Fixed"

The other feature where you may be affected adversely by dynamic dimensioning is in "Adjust spectral range" under "Operations"(3.1.3.3.10). But only if you are adjusting the range to a wider range than the current spectra, and there is not much reason to do that.

Chapter 6. Composing Figures and on-screen display

PostScript figures are made using (after selecting the traces to be included) the "Make Figures" button on the trace selection window (3.2.2.17), and the basic procedure is described there. The least-squares fitting routine also has the option to save a multi-page postscript file showing the goodness of fit. Both procedures can also create .pdf files from the postscript if you have GhostScript installed.

The option "Make PS Figure under the "disk-based" menu plots spectra from a series of binary spectrum file on the disk, rather than contents of memory. This lets you plot spectra when the number is too great to select in the Trace list (although it is questionable whether a figure with more than 35 spectra wold be intelligible). It is sort of a relic from the old QuickBasic program that held fewer spectra in memory, but it does allow some options like selecting the color and dash-pattern for each trace.

While the ""Make Figure" dialog allows to set the absorbance and wavelength scale, the easiest option is to compose the figure on the graphics screen and just click "use screen limits" buttons in the "make Figure" dialog. The absorbance and wavelength scales on screen can be set using "set limits" under the display menu((ctl-L>, 3.1.3.2.1), by autoscale on the trace list window (3.2.2.9), by using the "zoom and pan" buttons (3.1.2.8) under the graphics screen, or by dragging a rectangle on the current screen (3.1.1.1).

With a large number of spectra, especially if they have a specific sequence such as successive time points or spectra at different pH or potential; or successive fractions from chromatography, it is useful to vertically offset or "stack" the spectra (3.2.3.3.3), in order of their sequence, so they don't overlap and the changes occurring can be readily seen. If the spectra fall in groups that should all be plotted on the same baseline, but it is desired to separate the groups, select the spectra in a group and use the "add a constant" button on the trace selection window (3.2.2.18) to move them up or down. An example would be where multiple samples are analyzed but several spectra are taken on each sample, say after adding different reagents.

Either option actually changes the spectra in memory, so once you have the vertical arrangement you like, you can save the spectra as a new series in case you later want to tweak something and make the figure again.

Once you have offset spectra by varying amounts, the absolute absorbance scale printed by scanedit is no longer valid so you might want to erase the numbers (leave the tick-marks) and change the label to something like "Absorbance- 0.05/div", or put a vertical bar, the length of one or two divisions, and label it with the appropriate absorbance value.

If the set of spectra includes a baseline, it makes sense to subtract that from all before plotting, and perhaps not plot the baseline. Subtract from all is also useful for showing the changes taking place between spectra. If you want to show the changes taking place after adding a particular reagent, subtract the trace before that reagent was added, or perhaps a trace after the change is complete, from all the rest. If you decide to subtract a different spectrum from all, just do it- you don't need to reload the originals because the relative absorbance in the different spectra is preserved by the subtract from all operation.

If you are making the "decomposition figure" described in chapter 7, and you have two or three baseline polynomials among the fitting spectra, you can

reduce complexity by adding them and plotting only the result as "best linear (or quadratic) baseline correction" together with the actual fitting component spectra)

The "Make Figure" function (3.2.2.17) produces PostScript files, optionally also as .pdf if you have GhostScript installed(5.7). The PS files can be sent directly to a postscript printer, but ScanEdit does not provide for adding labels or other graphic features (other than labeling the axes and an optional label string over the top of the frame), so you will likely want to import them into a graphics program, either vector like Adobe Illustrator or a raster-based program like PhotoShop that can render postscript files. When you import to PhotoShop, you are asked to choose the pixel dimensions of the figure and resolution. The figures are nominally sized at 8.5 × 10 inches, and PhotoShop (my version at least) will suggest 612 × 720 pixels and 72 pixels per inch. (This is still the case even if you select landscape mode, then you have to rotate the figure after importing it.). If you change either number of pixels, PhotoShop changes the other to keep the aspect ratio the same. The important thing is the number of pixels, you can always increase the resolution after importing, by changing the resolution (without re-sampling) to make a smaller picture.

On importing a postscript figure, my version of PhotoShop shows it on a checkered background, which makes the fine traces hard to see. Do "flatten image" under the Layers menu to get a white background. If the Window is too small, PhotoShop shows it as a crudely resampled 66% size. Show it at 100% (or larger) to judge the quality of the image.

The colors for drawing the different traces come from a pallet of 16 colors (including black) which you can edit to suit your taste or making the traces easily distinguishable, using option "Edit PSPlot colors" (3.1.3.8.5) under the Preferences menu. If you edit and "accept" the colors, they will be effective for subsequent plots in the same session. If you "accept and save" they become permanent (see 3.1.3.8.5). The plotting program cycles through the colors starting with 1, in the order the traces appear in tracelist, but skipping over any unselected spectra- thus if you are plotting traces 1, 3, and 5, they will plot using colors 1, 2, and 3.

Finally, if you already have a figure that you are happy with except for the colors, you can edit the PS file to adjust the colors without having to remake the figure. Open the PS file in a text editor like notepad or MSWord, and search for "rgbcolor". This command will be preceded by three numbers setting the red, green, blue values, on a scale of 0 to 1, for one of the spectra. The first instance is for the first spectrum, and so on. You can edit the three numbers, save (without closing), and render in PhotoShop to see the result. repeat until you like it.

Chapter 5 says you can set the font to be used in labeling the axes, but that doesn't seem to be implemented yet. Coming soon! Now it is hard-coded to use Arial. If that is not available PhotoShop will probably use a suitable substitute. If you really need to change the font you can edit the PS file with a text editor. Find the line: "/hfont /Arial findfont 18 scalefont def" and change the name Arial to Helvetica or whatever. I believe the font name is case sensitive.

Chapter 7. Least squares fitting to a linear combination of standard spectra

7.1 Background

For solutions of chromophores that obey Beer's law, the spectrum of the solution is a linear combination of the spectra of the individual chromatophores; the amount of each depending on the concentration of that component in the solution. This has led to number of procedures to analyze mixtures when the spectra of the pure components are known.

7.1.1 Simultaneous equations/Matrix inversion: If absorbance is measured at a number of wavelengths equal to the number of components to be analyzed (and if the spectra of the components are sufficiently different at those wavelengths), you can set up set up simultaneous equations for the absorbance at each wavelength and solve for the concentration of each component. Applying to Warburg-Christian method of quantifying protein in the presence of nucleic acid[WC,kalckar],

$$A_{260} = [\text{protein}]\times E_{260,\text{protein}} + [\text{NucleicAcid}]\times E_{260,\text{NA}}$$
$$A_{280} = [\text{protein}]\times E_{280,\text{protein}} + [\text{NucleicAcid}]\times E_{280,\text{NA}}$$

for doing this by computer, it is convenient to express the simultaneous equations as a matrix equation, and solve by matrix inversion:

$$(A_i) = [E_{i,j}] \times (C_j)$$
$$(C_j) = [E_{i,j}]^{-1} \times (A_i)$$

where (A) is the vector of measured absorbance at different wavelengths i, [E] is a matrix of extinction coefficients of the different components j at the i^{th} wavelength, and (C) is the vector of concentrations of the different components j. Here it is assumed that the [E] is square, i and j have the same limits.

Papers have been published[WC,calckar,Arnon.Lichtenstein wiliams,etc] in a number of fields providing the extinction coefficients at selected wavelengths and deriving equations for the concentrations based on absorbances at those wavelengths.

This is a convenient method, but it is absolutely dependent on knowing that no other component in the sample absorbs at any of thee wavelengths and that the spectrum of each component in your solution, measured with your spectrophotometer, is the same as when the extinction coefficients ere determined. Presence of other absorbing components, failure of Beer's law (due perhaps to interaction between the components or with something else in your solution), or serious miscalibration of your instrument, will lead to inaccurate results with no warning sign (unless on or more concentration comes out significantly negative). You plug in the absorbance numbers, multiply by the inverse matrix, and you get the concentrations. If you calculate the absorbance at these points based on those concentrations, they will exactly account for the observed absorbances. But if you calculate whole spectra, the fit could be terrible. Matrix inversion gives an exact solution, even if it is nonsense. Like, "For

quality control, I slipped a tube of lime Kool-Aid into the batch of chlorophyll extracts that I sent to the lab for analysis. It came back as 2.37 μM chlorophyll A and 0.79 μM chlorophyll B! I need to find a new lab."

This method of simultaneous equations/matrix inversion also does not make maximal use of the available data. If you need absorbance at a number of wavelengths, the easiest procedure is probably to scan a spectrum and pick out the desired values. But that spectrum has information at so many more wavelengths! Surely the values at 279.8, 279.9, 280.1 and 280.2 are nearly as informative about the concentration of protein and nucleic acid as the value at 280.0. It would be nice to use the entire spectrum (which would also avoid the necessity of selecting the optimal wavelengths to measure, and of picking those wavelengths out of the spectrum for each sample analyzed.

7.1.2 Classical Least Squares-

This method uses a number of wavelengths greater than the number of components being analyzed, for example an entire spectral region in which the analytes' spectra can be distinguished. This ensures that those wavelengths with the greatest ability to distinguish the analytes will be included, and it can be shown that adding other wavelengths with less differentiating ability still contribute to improving the precision rather than diluting out the signal from the most useful wavelengths.

The problem is then overdetermined, and due to error ("noise"), the fit will not be exact. The least squares solution is the solution that minimizes the sum of square of the error at each wavelength (if necessary weighted for different noise levels at different wavelengths). Because of the gaussian shape of the normal probability curve, this is roughly equivalent to finding the combination of concentrations that would have the greatest probability of resulting in the observed spectrum. (In practice the probability aspect is relatively unimportant- the best fit "just fits" and any significantly different solution would give an impossibly different spectrum.)

In the least squares problem, the number of wavelengths is greater (often much greater, as in several thousand points in the spectrum and usually less than 20 components being analyzed), so the matrix of extinction coefficient $[E_{i,j}]$ is not square. There are J columns, one for each analyte, and each column is the spectrum of that analyte, with I points for the I different wavelengths. Being non-square, the matrix E cannot be inverted in the normal way, but a generalized inverse E^{-1} has been defined such that $E^{-1}E$ is the k×k identity matrix, and multiplying a spectrum by E^{-1} gives the least square solution set of concentrations. Thus the solution is formally similar to the matrix problem, only instead of multiplying the measured spectrum by a square inverse matrix, it is multiplied by a non-square generalized inverse matrix. In particular this matrix has J rows one for each component) that are I long (one element for each wavelength. These rows can be considered "probe" spectra, because taking the dot product of one with the measured spectrum gives the least-squares best estimate of the concentration of the corresponding analyte.

An alternative to the classic least squares method for analyzing spectra of mixtures is "reverse calibration", in which the probe spetr(a) are derived by fitting the "time courses" of the concentration of one or more compounds in a set of "training" spectra to the time courses at a number of wavelengths in spectra of

those samples. This can also be carried out in scanedit, using mainly options under the "disk-based" menu, but we have not yet made an automated routine for doing it.

7.2 The importance of including baseline function(s).

A small baseline offset , in the spectrum being analyzed or in one or more of the fitting spectra, can prevent a good fit and degrade the accuracy of the analysis. Dual-wavelength extinction coefficients are a classic way of getting around this: baseline changes contribute a constant or slowly -changing offset to spectra, so by choosing two wavelengths close together, their effect will largely cancel in the dual-wavelength measurement. The dual-wavelength extinction coefficient will generally be smaller than the single-wavelength one (unless the compound does not absorb at all at one of the wavelengths),which degrades the accuracy slightly, but knowing the baseline is not contributing significantly more than makes up for this.

In whole-spectral fitting of multiple components, the analogous device is to include one or more spectra simulating a baseline shift. This makes the orthogonal component of each spectrum smaller (effectively subtracting the mean of each spectrum from that spectrum) but if the spectra have significant variation over the range (peaks and troughs), what remains has all the information about the *shape* of the spectrum (as opposed to how high it reaches) needed to identify it in a mixture.

Bottom line- if you have absolutely clear solutions and you subtract a baseline made in the same solvent and cuvet, don't include baseline terms. Otherwise, or if you see evidence of baseline offset (best-fit and spectrum differing in regions of zero or low absorbance), use one or more baseline terms.

The program provides three levels of complexity in the fitted baseline: a constant offset over the whole spectrum, a straight-line offset increasing or decreasing linearly with wavelength, or a quadratic curve for baseline. That is achieved by including one, two or three baseline spectra in the list of fitting spectra included in the .mtx file of basis spectra. The first is a constant positive offset, the second is a straight slanted line with mean values zero, The third is a quadratic (parabola) centered at the center of the spectral range, with mean value zero and no net slant. These are constructed to be orthogonal, so we call them "orthogonal polynomials" but that may be a little too exciting a term for such a simple baseline correction. When you are preparing the spectra for inclusion in the fitting .mtx file, use the option "Add bsln polynom" under "operations" to add one, two, or three orthogonal polynomials.

7.3 How wide a spectral range should I use?

It might be argued, in favor of the few-wavelength methods like Avron or Warburg and Christian, that these few wavelengths can be chosen to maximize each component's difference from the other components. For example DNA (Nucleic acid) and protein are differentiated by their peaks at around 260 and 280 nm. In between the absorbances become more nearly the same, actually being the same at the crossover point. Will including these more ambiguous points dilute out the strong signal from the peaks, and just contribute their noise to the result?

It can be shown (empirically in this case, and a strong argument can be made theoretically for the general case) that this does not happen. The additional

points continue to add strength to the signal as long as one of the compounds has some absorbance. And if you are fitting the baseline, even the points where neither compound absorbs are helping, by defining the baseline and the shape of the spectra. So including the regions with characteristic absorbance features of each compound and perhaps a little more on the sides is good. This is discussed further in 7.8.

7.4 Importance of examining the fit.

One of the greatest advantages of fitting whole spectra as opposed to using only s many wavelengths as the number of components being analyzed, is the "goodness of fit" criterion. With equations based on the Warburg & Christian numbers for protein and DNA, or on the equations of Avron for chlorophyll A and B, you plug in the absorbance values and you get concentrations. If you take those concentrations and the extinction coefficients used in the equations, they will predict exactly the absorbances you entered. This will be the case even if the samples contained only red Easter-egg dye or lime Kool-Aid. However if you look at other wavelengths, they will be completely wrong. There is no goodness of fit test.

After scanedit calculates the best-fitting coefficients, it goes back to the original spectra and plots them alongside the best-fit constructed spectra and the residuals (error spectrum).If the fit is very good, the best-fit spectrum (cyan) overlays the original spectrum (yellow), masking it in most places. If the spectrum is noisy, the best-fit spectrum should make a smooth path through the noise, and residual should look like just noise..

When the spectrum is not fit well, examining the fit can give you an idea what is wrong. Is there a peak from an unexpected absorbing component showing up in part of the spectrum- Is the short-wavelength end trending upwards due to tail of an absorbance peak in the UV? If so, can you get a good fit by "omitting" the region in question?

Do the peaks seem to be shifted in wavelength, showing derivative signatures in the residuals? This could be to wavelength calibration error, or due to solvent effects affecting the position (and maybe peak width) of the spectral peaks. Are the regions of low absorbance also fit badly? This could be due to a baseline offset not being accounted for.

When each fit is displayed, the tracelist window is brought up, with the comment of the spectrum being fit and color key to the spectra (which is different from the usual first three colors, due to the need to make the best fit and original spectra highly visible and distinguishable). When the fit is poor, sometimes you may find yourself second-guessing it, wondering why it can't be improved by adding a little more or less of the component with a peak at such and such wavelength.

To better understand the fit, you can display a "decomposition" figure, decomposing the best fit into each of its components. Instead of hitting <enter> to advance to the next spectrum, enter f in the dialog box and hit enter. Now in addition to the original spectrum, best fit, and residual, you have each of the fitting components plotted, in the amount used in the best fit. The tracelist window shows the color key and comments for each of the fitting components, together with the numerical amount of each.

If you want to save these spectra to compose a decomposition figure for your notebook or for publication, enter a basename in the dialog box and hit

enter. The spectra in memory will be saved in a series with the name you entered. Otherwise just hit enter, and it will go on to the next spectrum.

7.5 Setting up for LSQ analysis

Basically you need one or more spectra to be analyzed, in the native binary format, forming a consecutive series if there are more than 1. And you need a set of spectra of the pure components being analyzed, the "basis" spectra. Ideally these should be at unit concentration, because then the results will read out in concentration in the same units.

The basis spectra should all have the same wavelength range and sampling interval, which you have already achieved if you have them all loaded in scanedit at one time. The wavelength range should be entirely included in the wavelength range of the spectra being analyzed but it can be a subset, shorter on either or both ends. The sampling interval must be the same as the spectra being analyzed.

So get the spectra loaded into scanedit (See chapter 4 for converting other formats). If the spectral range exceeds that of the spectra being analyzed, trim with "adjust spectral range" (3.1.3.3.10). You may want to order the spectra with those of more interest first (near the top). You probably want to add two baseline components (3.1.3.3.9), see discussion above(7.2). Then select all, or all that you want to include, and click the ".mtx" button under the "save" label near the bottom right of the file selection screen. This brings up a dialog where you can enter a name for the new file (extension .mtx is recommended). It is also an opportunity to restrict the wavelength range or sampling interval to that of the spectra to be fit. Changing the wavelength range may affect the orthogonality of the baseline spectra, but that is not a problem, and you can always load the resulting .mtx, make new baseline spectra, and save again.

Test the set of fitting spectra for linear independence by running the LSQ fit with the .mtx file you just made, with no spectra specified to be fit, and with the "show statistics" box checked. Interpretation of the statistics is described at the end of section 7.8

If this set of spectra will be used to analyze multiple datasets you may want to save it in your speclib directory instead of the current data directory. If you put a full path in filename textbox, the file will be saved there.

7.6 Other options in the LSQ fitting procedure.

The basic use is described in (3.1.3.6). Some of the options deserve more discussion.

If you check the box "save LSI matrix?" then the generalized ("least squares") inverse matrix used in the analysis will be saved in the file LSI.mtx. Loading this into scanedit gives the rows of the LSI matrix as spectra with the same number of points as the standard spectra in the fitting .mtx file, so you can load the standard spectra at the same time and verify the co-orthonormal relation between them using the statistics button (compare two traces) on the tracelist window. More practically, it is reasonable to believe that the extrema of the LS Inverse "probe" spectra would define the best wavelengths for monitoring the concentration, if a limited number of wavelengths must be used.

By default, the "autoscale" box is checked. This is convenient if you are fitting a number of spectra of different magnitudes: The plot showing the fit (and the plot showing the components of the fit, if requested) will be displayed on a

scale suitable for the spectrum being fit. The scale is adjusted for the whole spectrum, even if part of the spectrum is being excluded in fitting. If you prefer to set the scale before starting, and not have it changed, uncheck the box. This is good for fitting a single spectrum or a series of spectra of nearly the same scale. In either case, if you select to have a .pdf of the fitting results made, they will be on the same scale as the displayed fit.

7.7 Output files from the LSQfit procedure

The .csv file is described in 3.1.3.6

7.8 More insight into the least-squares fitting procedure.

If the Beer-Lambert law is applied at every wavelength to the spectrum of a solution with multiple absorbing species, we get an equation like:

$$A_i = \Sigma \varepsilon_{i,j} C_j$$

Where A_i is absorbance at the i^{th} wavelength, $\varepsilon_{i,j}$ is the extinction coefficient of the j^{th} absorbing species at the i^{th} wavelength, and Cj is the concentration of the j^{th} species. The summation is for j going from 1 to J, the umber of species being analyzed. This can be seen as a matrix multiplication, in which the row vector C_J is multiplied by the matrix E_{IJ} to result in the spectrum A_I, as a column vector.

$$A_I = E_{IJ}C_J$$

The columns of E are the standard spectra of the J components, each at unit concentration, and multiplying by C on the right makes the linear combination of these standard spectra that is the resulting spectrum A.

The problem being asked is to find the concentrations C, given spectrum A and the extinction coefficients in E. If the number of wavelengths measured I is the same as the number of absorbing species J, then the matrix is E is square. If in addition the spectra re all linearly independent, the matrix is invertible and a solution is given by:

$$C = E^{-1}A$$

This is mathematically equivalent to solving simultaneous equations as in the Arnon or Warburg-Christian methods, so it suffers from the problems mentioned at the beginning of the chapter- no "goodness of fit" criterion, and not using all the available data.

If the number of wavelengths is greater than the number of components, the matrix E_{IJ} is not square, its columns (the spectra) are longer than its rows. The problem is overdetermined, and in principle no choice of concentrations C will predict the spectrum A at every wavelength. In the absence of measurement error and with beers law followed exactly, we could pick any J of the I wavelengths to determine the concentrations by matrix inversion, and any such choice would give the same result. But taking noise into account, different choices of wavelengths would give different results, and some much less accurate than others. No solution would exactly predict the absorbance at every wavelength.

A generalized inverse of E is a matrix E^{-1} such that $E^{-1}E = I$, the identity matrix, with 1 at all the diagonal elements and zero elsewhere. If E is square, the generalized inverse is unique and is the inverse of E we mentioned above. If you think about how matrix multiplication works (the m,n element of the product is the m^{th} row of the left matrix dotted into the n^{th} column of the right matrix), saying that $E^{-1}E = I$ implies that E^{-1} has dimensions J,I; and that each row of E^{-1} gives 1

when dotted into the corresponding column of E, and zero when dotted into any other column. In other words, each row of E^{-1} is orthogonal to every column of E except the corresponding column.

Now remembering that the columns of E are the standard spectra, we can see the rows of E^{-1} as being "probe" spectra for each of the J compounds, that pick out the concentration of their compound and give zero for any other compound's spectrum, when applied by dotting into the spectrum in question. Then given the linear property of the dot product (the dot product of our probe spectrum into an experimental spectrum which is the sum of the spectra of the different components, is equal to the sum of the dot products with each of the components individually), this means that if we apply the probe spectrum to a mixture of the different species, it will respond only to the species for which it is the probe. And since it gives 1 when applied to the spectrum of that species at unit concentration, it will give the actual concentration (in the same units) of that species in the mixture.

We could describe the probe spectrum for a compound independently of matrix inversion, by defining it as a spectrum that gives 1 when dotted into the spectrum of its cognate compound, and gives zero when dotted into (i.e. is orthogonal to) the spectrum of every other compound that might be in the solution. The fact that $E^{-1}E = I$ shows that the J rows of E^{-1} are probe spectra fitting that definition for each of the J compounds whose standard spectra are the columns of E.

The generalized inverse of E is not unique if the number of wavelengths I is greater than the number J of compounds. However there is a unique choice of generalized inverse that is statistically superior to the others. We call this the least-squares inverse matrix. because it gives the solution that minimizes the sum of squared error in predicting the observed absorbance at each wavelength.

For example, with 1000 WL points, we are in 1000-D space (R^{1000}). If we take our probe spectrum for cpd 1. This is orthogonal; to cpds 2,3,4,5 but not orth to 1, since it fits the definition of a probe for cpd 1. Now if we take the component of that that is orthogonal to each of a dozen other compounds that are not in the set being analyzed, it will still be orthogonal to cpds 2-5 and not to 1, so still a generalized inverse. However whenever you take the component of a spectrum orthogonal to another spectrum, the norm of the spectrum becomes shorter. This follows from Pythagorean theorem.

Another way to get alternate lsqinv is to set A at some wavelengths in the standard spectrum to zero before making the LSQI. This results in those wavelengths not being used, the corresponding wavelengths will be zero in the probe spectra as well. . But again, this is making the length of the orthogonal component of the standard spectrum sorter, and the result will not be as precise.

The relation between length of the orthogonal component and precision of the measurement is given by a quantity we call the "standard deviation multiplier" or SDM. Given that the concentration is calculated as a dot product between the spectrum and the probe spectrum, application of the rules of error propagation show that the standard deviation of the calculated concentration is equal to the square root of the sum of the squared products of the elements of the probe spectrum and the standard deviation at corresponding wavelengths. With the simplifying approximation that the standard deviation of absorbance is the same at all wavelengths, its square factors out of the summation, and the standard

deviation of the calculated concentration is the standard deviation of absorbance measurement times the square root of the sum of squares of elements of the probe spectrum. This latter quantity is what mathematicians call the norm, or length, of the probe spectrum. In the case of probe spectra, we call it the standard deviation multiplier (Berry & Trumpower), as it multiplies the standard deviation of the absorbance measurement. Clearly the smaller the SDM, the more precise the measurement of concentration will be. Other papers have independently come up with the same quantity, called ?? (??) and ac60226a023.pdf?,

But how is this related to the length of the orthogonal component, and why did we say that should be larger to give more precise results? This is due to the "standardization" of the probe spectrum. Say we have a component of the standard spectrum of species j that is orthogonal to all the other species being analyzed. Used as a probe spectrum, that would give zero in response to all the other spectra, but it would not in general give 1 in response to the corresponding spectrum, and hence would not give the correct concentration of compound j. To standardize it, first we normalize it by dividing by its length. This gives a unit length vector in the orthogonal direction, and dotting the standard spectrum into gives again the norm of the orthogonal component. In order for it to give 1 we must divide this unit-length vector again by the norm of the orthogonal component, giving a proper probe spectrum with length equal to the reciprocal of the length of the normal component.

Thus the SDM is the reciprocal of the length of the orthogonal component of the standard spectrum. In fact we never go through these steps, but by definition ($E^{-1}E = I$) the rows of a generalized inverse matrix give 1 when dotted into the corresponding spectra, so we know their length is the reciprocal of the components of those spectra along the rows of the inverse.

Above we specified "an orthogonal component", and "a generalized inverse", acknowledging these are not unique when the number of wavelengths is greater than the number of spectra being considered. However the specific least squares inverse matrix has rows that are along *the* orthogonal components of the standard spectra. Any other orthogonal component will be related to *the* orthogonal component as one leg of a right triangle whose hypotenuse is the orthogonal component, and so shorter, and their SDM will be greater. (By *the* orthogonal component we mean the component of the standard spectrum that is orthogonal to all the other spectra, that is the standard spectrum is equal to the sum of the orthogonal component and the component within the span of the other spectra. When the number of wavelengths is greater than the number of standard spectra, the orthogonal subspace has dimension greater than one so other orthogonal components exist, but *the* orthogonal component is the longest).

Another statistic is the "linear independence". The linearly independent" component of the standard spectrum is just this orthogonal component. We define linear independence as the norm of this orthogonal component divided by the norm of the standard spectrum, i.e. how much of it is left after subtracting out the components along each other spectrum. Obviously if two spectra are very similar, their orthogonal components will be much smaller, and linear independence is low.

However a small linear independence is not necessarily worrisome if the standard spectrum is quite large, the orthogonal component may be a small component of the standard spectrum but still large enough for precise

measurement. This can happen when the standard spectrum has rather large absorbance everywhere in the range, but still has a distinctive shape, and one of the other components is a constant baseline as recommended (7.2). In order to be orthogonal to the constant baseline, the spectrum drops down to where its average value is zero, so its norm is greatly decreased. But due to the peaks and troughs, it still has enough shape to be precisely determined.

The other thing reported by LSQ-fit reports if you ask for statistics, is the cross-correlation matrix (Not Pearson correlation, the means are not subtracted). This is the matrix of pair-wise dot-products of the spectra after normalization, which is equal to the cosine of the angle between them in multidimensional space. The diagonal elements (self-correlation) are all one and the off-diagonal elements range from zero to one. Value close to 1 are worrisome, but not necessarily problematic. Even a cosine of 0.99 corresponds to a sine of 0.14, meaning that 14% of the length of the standard spectrum remains after taking the component orthogonal to the other spectrum.

The final criterion is the SDM, but even here things may not be as bad as they seem- you have to consider the units. The algorithm assumes the SD of absorbance is in absorbance units (or whatever units your spectra are in), so a reasonable SD would be 10^{-4} to 10^{-3}, and if concentrations are going to be in 10's or 100's, a pretty large SDM would still correspond to a small percent error. The calculated concentrations are in the units of the standard spectra. So if the standard spectra are 1 µM, but you are measuring concentrations in the mM range, again the percent error could be small, Of course the concentrations in your sample should be such to fill a reasonable part of the linear range of the spectrophotometer, say between 0.2 and 2 absorbance.

If the spectra of two components are really almost the same, it may be useful to replace them with their sum and difference spectra, which span the same space. Then you will get a small SDM and high accuracy for the sum of their concentrations, and a less accurate measure of the difference which tells you how much is each component

Linear regression in excel:
for one independent variable (y=mx+b):
no stats:
select block of two cells for the results
enter "=LINEST(g6:g13,F6:f13)" - you can drag to select the ranges
<ctrl><shift><enter>

with stats:
select block (of two cells wide by 5 high) for the results
enter "=LINEST(g6:g13,F6:f13,true,true)" - you can drag to select the ranges
<ctrl><shift><enter>
First row is your results as before
second row is standard error of each param
third row is CC, standard error of the Y estimate

For more than one indep variable, just select wider block and more columns.

Chapter 8 - using SVD

8.1 Background

Digital spectra are sampled at a finite number of discrete wavelength points, and hence are "ordered n-tuples" and vectors. A spectrum with say 4001 data points is a vector in 4001-dimensional vector space. However if a dataset contains only 50 such spectra, those spectra span at most a 50-dimensional subspace of 4001-D space. Furthermore if these 50 spectra all contain different mixtures of the same seven light-absorbing compounds, and Beer-Lambert law holds, then each spectrum is within the error level of the span of the spectra of those seven compounds.

Singular value decomposition is a way to reduce the dimension of a dataset by expressing the experimental spectra as linear combinations of a number of spectra, which we will call the "singular spectra". Assuming there are more wavelengths than spectra, SVD reduces the dimension to the number of spectra with no loss of information.

SVD also serves to segregate noise from real features of the spectra. The real features, occurring in multiple of the experimental spectra, tend to be represented in the singular spectra with the largest singular values, while the random noise features, unique to each spectra, are relegated to the singular spectra with the smaller singular values. This allows a further reduction in dimension, losing some of the uncorrelated noise but keeping the recurrent features of the spectra in the dataset.

The procedure "decomposes" a matrix containing a number of spectra into three other matrices:

$$A_{i,k} = U_{i,h}\, \Sigma_{h,h}\, V^{T}_{h,k}$$

Here A is a matrix containing the absorbance data. The columns of A are the spectra, indexed by row index i, the wavelength index. Each row of A corresponds to one wavelength in all the spectra, indexed by column index (spectrum number) k. We refer to the rows as "time courses" at the different wavelengths, because if the different spectra were taken at successive times, say during a kinetics experiment, each row is the time course of the absorbance at that wavelength during the experiment. Other terms might be more appropriate if the independent variable between the spectra is pH, Eh, elution volume, or just spectrum number, but for consistency we call the rows "timecourse" in any case.

U, Σ, and V are the outputs of the procedure. The row index of U is i, the wavelength index, so the columns of U are vectors in the same vector space as the columns of A, the original spectra. We call The columns of U, formally known as the left singular vectors, the "singular spectra". Multiplying a matrix on the right makes linear combinations of the columns of the matrix, so the fact that multiplying U by $\sum V^T$ on the right gives A, means that each column of A (each original spectrum) is a linear combination of the columns of U, the singular spectra. So the singular spectra span the same subspace of N-dimensional space in which our experimental spectra exist.

The matrix V^T has column index k, like A, so the rows of V^T (columns of V)

are in the same vector space as the rows of A, the timecourses. We call these (the right singular vectors) the singular timecourses. Multiplying a matrix on the left makes linear combinations of the rows of the matrix, so the fact that multiplying V^T by $U\sum$ on the left gives A means that each row of A (original time course at each wavelength) is a linear combination of the rows of V, the singular timecourses. So the singular time courses span the same space in which our experimental timecourses exist.

The index h is the singular component number. It is the column index of U, and the row index of V^T, so it gives the number of singular spectra and singular time courses. And they are strictly linked: the elements of the h^{th} time course give (after multiplication by the constant weighting factor $\sum_{hh}$) the amount of the h^{th} singular spectrum in each of the spectra in A.

It may be helpful to compare the singular value decomposition to the decomposition defined by Beer-Lambert law, which is familiar to most chemists and biologists. The law states the $A = \varepsilon C$. If we are measuring at multiple wavelengths and more than one compound in the solution absorbs, the contribution of the j^{th} compound at the i^{th} wavelength is given by:

$A_i = \varepsilon_{ij} C_j$, the extinction coefficient of the jth compound at the ith wavelength times the concentration of the jth compound. If we have a dataset with more than one spectrum, it becomes the matrix equation

$A_{ik} = E_{ij} C_{jk}$

A is the same as in the SVD, columns are the experimental spectra. E is the matrix of extinction coefficients, the columns are the spectra of the pure compounds at unit concentration. C gives the concentration of each component in each spectrum: Thr k^{th} column gives the concentration of each compound in the k^{th} spectrum The j^{th} row gives the timecourse of the j^{th} compound in the experiment. By analogy to E, in SVD U contains the spectra of the singular components, and $\sum V^T$ gives the amount of each of those spectra in each of the experimental spectra.

$A_{ik} = U_{ih} [\sum V^T]_{hk}$

The singular spectra form an orthonormal set, as do the singular timecourses. That means each is orthogonal to all the others in the set, and that each has "norm" (a kind of magnitude, = square root of the sum of the square of the elements) equal to 1. So all the spectra are about the same "size", as are all the timecourses.

But we said some singular spectra contribute more than others. This is managed by the matrix $\sum$. It is a square diagonal matrix, the diagonal elements being the weighting factors: Σ_{hh} is the weighting factor for the h^{th} singular spectrum or the h^{th} singular timecourse. In fact we apply $\sum$ to the time courses, multiplying V^T by $\sum$ on the left before saving it in the <prefix>.cmp file. This is what you want for doing a global fit of the singular time courses to a model for how the concentrations vary. You wouldn't want the time course of the last singular component, which makes a miniscule contribution to the data set and consists mainly of noise, to be weighted equally with the stronger components.

In fact if some of the singular components contribute almost nothing and consist mainly of noise, we may want to skip them. This can be tested by reconstituting the spectra with different numbers of the strongest components. We re-order U, Σ, and V in order of decreasing singular value. There is an option

to reconstitute the original spectra using the first m components, and calculate the rms error. This is useful for determining the real dimension of a dataset. If only 3 absorbing species are present, three singular components should reproduce the spectra with a high degree of accuracy. It is common to say things like "three principal components reproduced 98% of the spectral signal". You can also look at the list of eigenvalues printed when the procedure runs, and saved in <prefix>.evl (these are actually the squares of the singular values). The magnitude decreases for the first several values, then levels off at the noise level. This can give you some idea of how many components you will need.

The singular time courses are the eigenvectors of the cross-correlation matrix A^TA, and the singular values are the square roots of the corresponding eigenvalues. The eigenvalues are printed on the screen during the run and listed in the file <prefix>.evl. The singular spectra in U can be obtained as the eigenvectors of the outer correlation matrix AA^T, but once we have the singular timecourses and values, it is more convenient to calculate them from the definition of SVD, which is what we do:

$A_{ik} = U_{ih} [\sum V^T]_{hk}$ Note that V^{T-1} is just V, since the rows are orthonormal

$A_{ik}[\sum V^T]^{-1} = U_{ih}$ $[\sum^{-1}]_{hh}$ is just $1/[\sum]_{hh}$ for every h.

This amounts to taking the h^{th} time course and dotting it into the time course at every wavelength in A to determine the contribution of the h^{th} singular spectrum at that wavelength.

A global fitting experiment involves fitting the absorbance at every wavelength in every spectrum of the data set to a model. For example for a potentiometric titration monitored by whole spectra, the model would include midpoint potentials and "n" values of each redox-active component, as well as extinction coefficients (or difference extinction coefficients) of all the components. At each wavelength the "time course" of absorbance will be a linear combination of the timecourses of the different components (depending on their extinction coefficients), so at every wavelength, the "time course" of absorbance is fitted to a sum of Nernstian curves.

Since the time courses are vectors in 100-dimensional space and there are thousands of them, they cannot all be linearly independent- all could be represented as linear combinations of 100 basis spectra. And that is what SVD does, reducing the number of time courses from one at each wavelength (I in the above equation) to just the same number as the number of spectra. (K). Furthermore if there are only 5 titrating components, the actual dimension of the space is only 5, and all of the relevant information is in the first 5 singular components. There is a one-to-one relation between fits to the singular timecourses and fits to the absorbance at every wavelength, and the best fit to one will correspond to the best fit to the other, provided the time courses are weighted by their singular values. This is why the time courses saved in the .cmp file are first multiplied by their singular values.

The fit will provide a list of Eh for each transition, and the amplitude of the change in each of the singular time courses fitting that transition. This tells the linear combination required to regenerate the time course or spectrum of the titrating components, by combining the singular time courses or singular spectra in these proportions. A good test is to generate the spectra (which is what "Real spectra from eigenspectra" does, lsqfit to fit the original spectra to these derived spectra. The resulting concentrations of each compound should follow a smooth, single transition.

8.2 Procedure:
The SVD functions are accessed from the SVD drop-down menu on the graphics window (3.1.3.4).

8.2.1 SVD.

The first Item, "SVD" performs the SVD. The files to be processed should be in a series, that is they all have the same basename and consecutive 3-digit extensions. And the number of spectra should be less than or equal to the number of wavelength points in each. (We can provide a routine for the case where the number of spectra is greater than the number of wavelengths, but it has not been incorporated yet)

Click "Browse" to select the first spectrum and either enter the number of the last spectrum or click "Last" to browse for it. Also enter a short string for a filename prefix to identify the files produced in this run (or leave the pre-filled "abc"). Then click "Begin".

If you selected a file that is not in the current directory, you are asked if the results should be saved with that file or in the current directory. If you select yes, the data directory is set to that directory and results are saved there.

Several msgboxes pop up telling you what is going on: completed correlation matrix, finding eigenvectors by Jacobi, Jacobi completed with largest offdiagonal = ?", "We will now construct the singular spectra". Read the messages, then click OK or press <enter>, to continue.

The singular spectra are plotted one after the other as they are calculated, with a short pause to let you see them. They are also saved in a series of binary spectra named <prefix>SV.###. After the first few spectra they start to look like random white noise. This indicates that your spectra are nearly accommodated in a much smaller-dimension vector space, probably due to the limited number of chromophores.

After construction of the singular spectra, the SVD operation is finished. But the program carries out two tests to ensure the operation was successful. These were mainly for debugging, and should be made optional now, but anyway they don't take much time.

Test Eigenvalue Decomposition- This is the same as the option below with the same name (8.2.2), except it is set up automatically with information from the singular value decomposition just preformed.

Test Singular Value Decomposition- This is the same as option (8.2.3) below: Reconstitute Orig. Spectra w N components. In this case it uses all components, which should reproduce the spectra exactly to within the precision of the arithmetic.

Output files: <prefix> is the string "abc" or whatever you changed it to in the initial dialog.

<prefix>.evl - A list of the eigenvalues of the correlation matrix, sorted in decreasing order. The values are listed one to a line, preceded by a line with "1" and a line with the number of eigenvalues (equal to the number of spectra)

<prefix>.dot The square "cross-correlation" matrix. The i,jth element is the dot product of the ith and jth orig spectra.

<prefix>.csv - The singular timecourses in spreadsheet format. The timecourses are the columns, with one additional column inserted in first place containing the spectrum number. The singular timecourses are already multiplied by the singular values to give evenly-weighted values for fitting.
<prefix>.cmp - The same information, in a form some of our programs can read
<prefix>.stt - Some statistics. Probably for debugging, not useful.
<prefix>SV.### - the singular spectra, binary format
<prefix>RC.### - the reconstituted spectra produced by the second test, binary format

8.2.2 Test EVD (eigenvalue decomposition)

Mainly for deugging and confidence-building. Regenerates the correlation matrix from its eigenvectors and eigenvalues. It will report the RMS Fractional Error, or difference between the original and reconstituted matrix, which should be on the order of 10^{-11}.

Selecting this you are asked to find the .evl file from the decomposition being tested. This gives it the prefix and path of the SVD output files, which are all it needs for the test.

8.2.3 Reconstruct Orig. Spectra with N Components -

This tests the overall SVD by regenerating the original spectra from their singular spectra, singular timecourses, and singular values. Useful for exploring the dimension of the spectral dataset, related to the number of independent absorbing compounds.

By default it uses all singular components with non-zero eigenvalue, and it reproduces the spectra, including the noise, exactly to within the precision of the math. Since the last eigencomponents (with smallest singular value) consist mainly of noise, you can do the reconstruction with fewer components.

Selecting this brings up the dialog in Figure 8.2, similar to the dialog for SVD. The top box is for the filename "prefix" that was used during the decomposition. Even if it is prefilled with the right prefix, you should fill it again by clicking "choose .evl file" to browse for that file, because this will provide the pathname to the directory where the output files are, in case it is not the current directory. Then click the Browse button to browse for the first experimental spectrum. This fills both path and filename for the first spectrum. It is not necessary to specify the last, because the number of spectra is equal to the dimension of the .dot file.

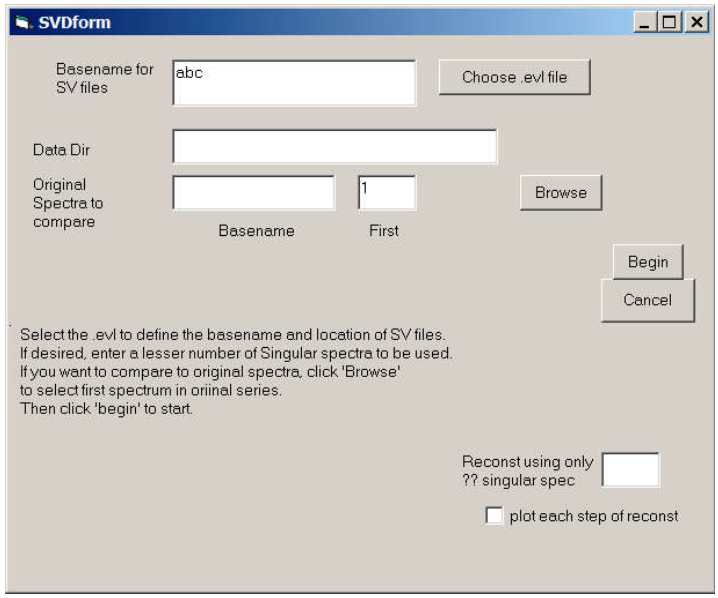


Figure 8.2- "Reconstruct" dialog box.

If you want to limit the number of components used in the reconstruction, enter the number in the box labeled "Reconst using only ?? singular spectra". If this is blank then

all spectra (with non-zero singular value) will be used. Then hit begin.

For each experimental spectrum, it plots the original spectrum in blue, the reconstituted spectrum in green, and the difference in red. Text below the graphics tells you which spectrum this is and how many components were used in the reconstruction. Then it halts with a msgbox to give you time to examine. Hit enter (or click OK) to go on to the next. If you type "q" before hitting enter (replacing "ok" with Q in the msgbox) it will halt the construction of spectra and print statistics for the traces so far.

When finished, it will report the overall RMS error, over all the wavelengths and all the experimental spectra. This will be in the units of the original spectra, normally absorbance. Using all components, this will be a ridiculously small number like 1×10^{-10}, indicating it is fitting the traces, including the noise, exactly. As you decrease the number of components this statistic will increase to something like the actual error in the absorbance measurement, probably on the order of 10^{-3} or 10^{-4}. If it gets much higher than that, you need to use more components to account for all the real variation in the spectra.

8.2.4 Real Spectra from fit to Singular timecourses.

As mentioned in "background", the singular component act like mythical compounds, that have their own standard spectra and are present at different concentrations in the different experimental spectra. By reducing the number of singular components to the minimum required to adequately reconstitute the experimental spectra, you can get an idea of how many actual species are varying. But the singular spectra are not the spectra of the real compounds, they are different linear combinations of them. The time courses of the singular components are not the time courses of the real compounds, but linear combinations of them. Suppose you take spectra of a solution at different pH values, and the solution has ionizable components at three well-separated pKas that affect result in a spectral change. If you could plot concentration of the ionized form of each, vs pH value, they would each show a single transition with predictable shape at the pKa of their ionization.

If you plot absorbance at a wavelength affected by all three ionizations, It would show steps at all three pKa's. By fitting this curve to a Henderson-Hasselback equation with three terms, you could get estimates for the pKa's and for the absorbance change at each wavelength, you would obtain the spectral changes at each wavelength, the difference spectra associated with the three ionizations. If you make the fit global, fitting all wavelengths simultaneously with the same three pKa', you would get a more accurate estimate of the pKs and the difference spectra.

In this way the singular timecourses behave like the timecourse of absorbance at a single wavelength. Since they are linear combinations of the time courses of the real components, they will show steps at all three pKas, and can be fit to the same H.H equation. From the relative amount of each component changing at each pK, the relative amount of the different singular spectra contributing to that change can be known. By combining the singular spectra in those proportion, the full spectra of each real compound can be generated.

Of course this can be done manually with the "linear combination" operation, but if the parameters are determined by a fitting program and written

to a file, it is convenient for scanedit to be able to read that file and generate the spectra.

On choosing this option, you are asked to find the .fit file, using the file-open dialog. This refers to a file that contains the fitting parameters, especially the amount of each singular component corresponding to each real component in the model.

The format is basically a table, with one row for each real chemical component and one column for each parameter determined by the fitting program. These are preceded by three lines containing: the number 2 (dimension of the table), the number of rows, and the number of columns.

In each row, the parameters are listed in the same order: first one or two nonlinear parameters (which are important for the fit but not for generating the real spectra) and then the fitting amount of each singular component used.

The first line is the number 2, the second is the number of components in the model (pKas in the example above). The third line is the number of parameters used to describe each component- this is the number of singular components used in the fitting plus one or two nonlinear parameters.

The following lines (one for each chemical component, simply list the parameters- one or two nonlinear parameters (halftime, pKa, Em +n-values) which are ignored, followed by the amount of each singular component contributing to that chemical component. If you are setting up this file by hand you still need to put one or two dummy values on each line, and you will be asked how many. But if you are setting this up by hand, it would be easier to make the linear combinations if singular spectra using the linear combination operation on the graphics screen menu.

Although we describe this as a table, the inputting routine uses spaces, commas, or newlines as delimiters, so the numbers can be one to a line- only the order matters. An example of a fit file for fitting three singular values to four-component Nernstian curve is shown:

```
 2
 5
 5
0         0   -2.605862E-03   2.403427E-03   2.059955E-03
775.3784  1    3.022723E-02  -0.1104711     -4.822008E-02
651.4222  1    6.672915E-02   9.67249E-03    2.533652E-02
553.6945  1    0.1118848      2.884065E-02   2.464763E-02
434.8021  1    9.289244E-02   5.447253E-02  -3.315383E-02
(This is formatted as a table- the actual file had one number per line,
i.e. newlines for separators.)
```

This fitting program generates 5 components for a 4- component fit- the first is the fully oxidized absolute spectrum, and the other four are the spectral changes incurred by each titrating component. So there are 5 rows. each row contains the midpoint potential and n-value (0 and 0 for the oxidized spectrum), and the amount of each singular time course titrating at that midpoint potential.

After giving it the .fit file, you are asked whether this is potentiometric fit. Say yes if there are two nonlinear parameters before the amounts of singular components on each line; no if only one.

Then you are asked to find the eigenvalue file resulting from the SVD. Then the fit file is opened and the number of singular components used in the fit is determined. You are given the opportunity to use fewer, but there is probably no reason you would want to. Finally, you are asked to give a basename for the

series of binary spectra to be produced It makes the specified linear combinations and saves them in a series with that basename.

8.2.5 Test data for SVD

If you would like to test the SVD and resconstruction procedures on multi-wavelength, multicomponent spectra, look in the "tutorial" packge (appendix A) and take the spectra under demo-1 or demo-2 folders. The former contains spectra of the mitochondrial bc1 complex, a mult-cytochrome protein complex, in different redox states adjusted by adding small amounts of chemical oxidant and reductant. The latter contains spectra of succssive fractions in chromatography of photosynthetic pigments.

Chapter 9. The Fourier menu:

The Fourier transform allows you to decompose your spectrum into a series of sine waves of different frequencies and amplitudes, and regenerate the spectrum after perhaps modulating the amplitudes. By selectively attenuating the different frequency components you can implement a low-pass or high-pass filter, the former being useful for removing high-frequency noise. And it provides a neat way to "resample" the spectrum, changing the sampling interval. For example, if your spectrum is sampled at 10 points per nm, you can easily save a subset at 5 ppnm or 2 ppnm using the "save ascii" option from the console. But if you need 25,or 8, or 4 ppnm, you don't have those points so interpolation is necessary. The inverse FT can be evaluated at arbitrary points within the spectrum to provide whatever sampling you need.

Note that the "frequency" and "wavelength" discussed here have nothing to do with the wavelength of the light used to take the spectrum, they have to do with how smooth or how jittery the spectral curve is. However since the X-axis on that curve is labeled "nm", the units will be in nm or nm^{-1}.

Menu Items:

9.1. Fourier transform. This calculates the FT of a spectrum in memory, saving it in the next available trace. Some mathematical details are given at the end of this section.

The first dialog asks for the number of the trace to be transformed. You need to have already loaded it into memory.

Once you dismiss the second box the calculation proceeds, and when it finishes the transform is plotted and a message pops up telling you it is done. For normal spectra the transform has large positive and negative values in the first few points, decaying to a line near zero. for most of the trace. This is because visible/UV spectra tend to be smooth, with little high-frequency component. You can save the transform like any other trace to reload it later, but normally it is just generated in memory and used right away.

9.2. Inverse FT. Given the transform calculated with (9.1) above, back transform to regenerate the spectrum, with or without limiting the frequency range to use.

The first box asks which trace to back-transform. If the trace was produced by the Fourier transform procedure above, it will have a flag identifying it as such. If this flag is not present you will be warned ("This does not seem to be a Fourier transform") but you can go ahead and use it anyway if this is what you want. This allows experimentation with different artificial transforms, and verification that the procedure does what it should in specific simple cases.

The next dialog allows you to limit the high and low resolution to use. The cutoffs can be expressed as frequency (cycles/nm) or period (nm) or simply the number of Fourier terms. The cut-offs can be specified as the highest order to include, the highest frequency (cycle/nm) or period (nm). The dialog comes up set to use all orders from zero to one less than the number of points. This latter corresponds to just below the Nyquist frequency, which is 1 per two sample intervals.

If you set period to 0 for the lower limit, it instead gives frequency 0. For the high limit, setting period to zero gives the highest frequency below the Nyquist limit. If you change anything, hit the "Verify" button to recalculate exact frequency and period for the integral order number, and to verify you are below the Nyquist frequency. Then press "continue".

This is a simple cutoff - truncating the series and putting 0 for all higher (or lower) frequency coefficients. It asks for the period in nm* of the highest frequency to include- so if you want to remove all frequencies higher than 2 per nm, you would put 0.5. To remove all frequencies higher than 1 per 5 nm, put 5. The high-frequency cut-off comes prefilled with a period twice the sampling interval, which calculates the full transform up to just lower than 1 per 2 sampling points- 5 per nm if you have 10 points per nm. This is the Nyquist limit.

Truncating the series like this can cause Fourier ripples if the function being transformed still has significant frequency components beyond your cut-off resolution. To avoid ripples it may be better to use the damping procedure below. With that the original spectrum should be reproduced exactly, with all its noise. However UV-visible spectra tend to be rather smooth - partly due to the uncertainty principle. The peaks in the cytochrome spectra we are working with tend to have bandwidths about 15-20 nm, and truncating at 5 or even 10 nm causes very little distortion of the peak shapes.

Note that although we use nm here as the units of the axis, the could be anything depending on what your spectra are. In one case we analyzed diffraction powder patterns as spectra, the X-axis was 2-theta angle in units of degrees. If the X-axis of your spectra is not in units of nm, substitute for nm whatever the units plotted on the X axis are.

The effect of truncating the series is seen in Figure 9.1. The Fourier transform of a spectrum (of cytochrome c plus ascorbate) was back transformed without truncation (green) and with truncation at frequencies corresponding to 5(red), 10 (cyan), or 20 (magenta), nm. The original spectrum is plotted blue, but completely overwritten by the UN-truncated spectrum (green) which in turn is largely obscured by the red (5 nm) and cyan (10 nm) traces. The 20 nm truncation (magenta) shows severe distortion, especially at the peaks. Notably the peak at 266 nm (due to ascorbate),which is wider than the cytochrome peaks, is much less distorted. To better quantitate the distortion, The original spectrum was subtracted from all and the differences were plotted on the same scale (lower traces in upper figure) and 4 × magnified (lower figure).

The untruncated transform (Frequency 5/nm, period 0.2 nm) exactly duplicates the original. With period 5 nm, no difference is noticeable in the plot, but the difference spectrum shows slight ripples. At 10 nm these become more noticeable and the difference can be seen without subtracting. With 20 nm the spectrum is severely distorted. You may need to enlarge the picture on your screen to see the distortion with 10 or 5 nm truncation.

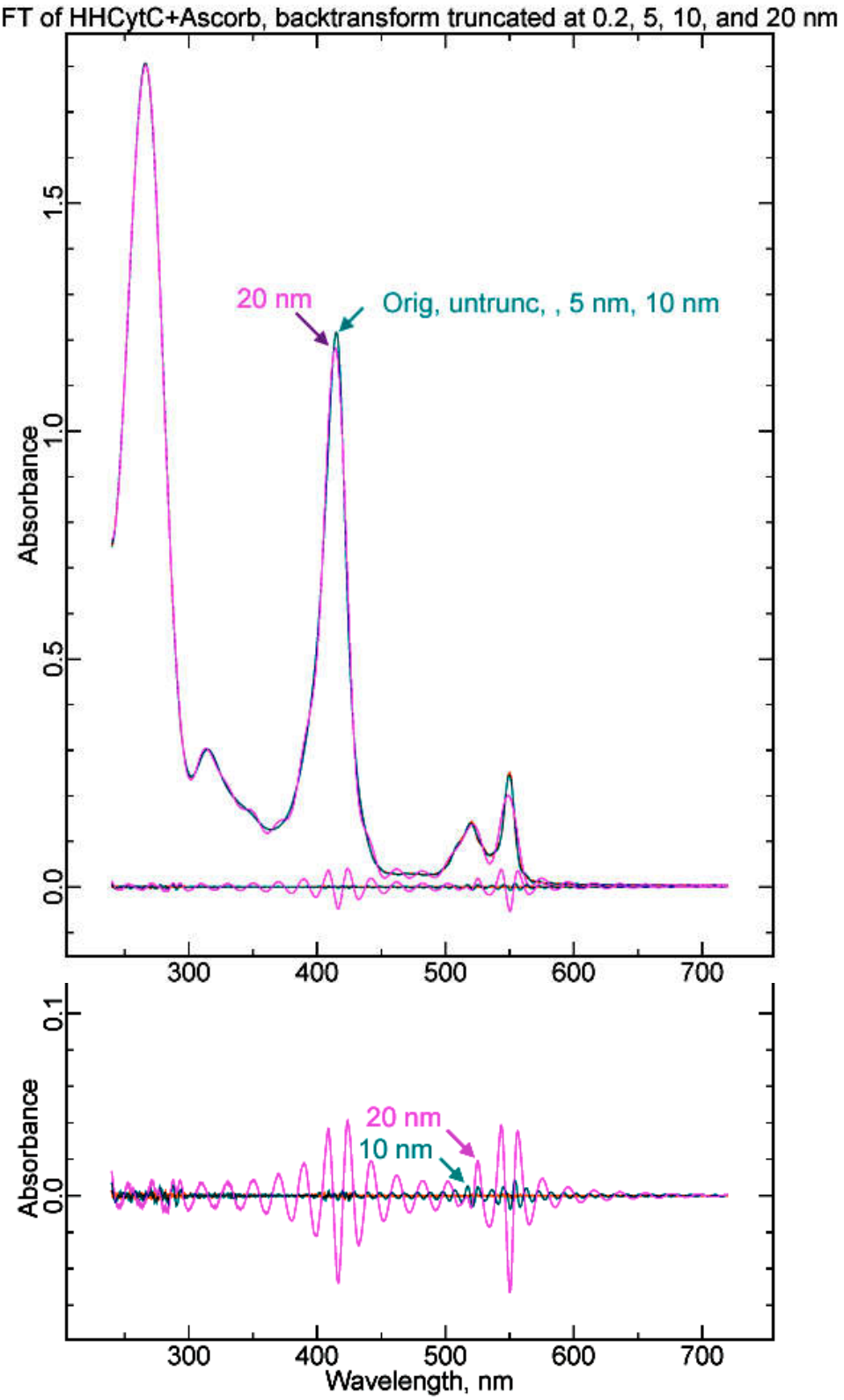


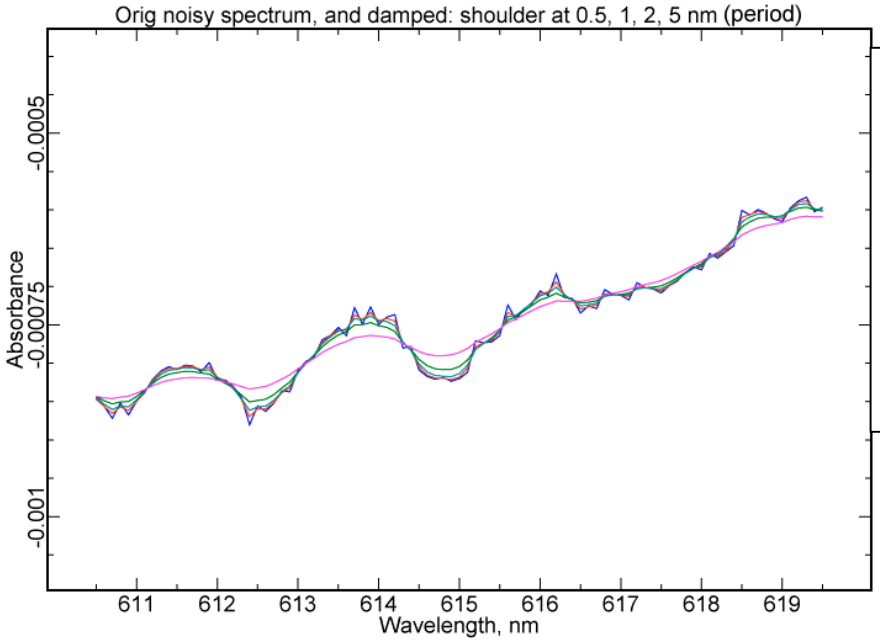
Figure 9.1. Effect of truncating a Fourier transform. Above: original and reconstructed spectra. Below, differences between original and reconstructed spectra truncated at a frequency corresponding to a period of 0.2, 5, 10, and 20 nm.

9.3. Damp a transform. "Damping" is achieved by attenuating each coefficient of the transform by an amount dependent on frequency, after the pattern of the Bode plot for an RC low-pass filter: damping factor = 1/sqrt(1+(Fs/F)^2) where F is the frequency and Fs is the "shoulder" frequency at which the damping factor is 1/√2. On a log-log plot this approximates a constant amplitude at frequencies below the shoulder and falling off by 10-fold per decade at frequencies above the shoulder.

This routine simply multiplies each coefficient of the transform by the appropriate factor and saves it in a new transform in the next available trace. The next function applies the damping factor while doing a back-transform and saves the filtered spectrum in the next available trace. Why would you want to apply the damping factor without doing the back-transform? Maybe to apply the damping multiple times for sharper cut-off, or to prepare for a back-transform that does not support damping (inverse FT with resampling, below).

The first box asks which trace to damp, and the second asks for the period (in nm) for the shoulder frequency. Again, a value of 0 for the shoulder period (infinite frequency in principle) puts the shoulder at a wavelength 2 times the sampling interval. This will result in coefficients at the highest frequency (Nyquist limit) being damped by a factor of √2, and is a good starting point for a minimally distorting smoothing. Depending on how sharp your spectral features are, you may be able to go up to 1 or 2 nm without significant distortion. Entering a wavelength less than 2 times the sampling interval is supported, and will give even less smoothing/distortion.

9.4. Damped inverse FT. This is the same as 3 except that instead of saving the modified coefficients it uses them to do a back-transform and puts the result in the next available trace. The comment specifies what shoulder frequency was used. This is the fastest way to compare the results with different shoulder frequencies to find the optimum tradeoff between spectrum distortion and remaining noise.



The absorption scale here is not correct, because the spectrum had been standardized by multiplying by a small number.

9.5- Frequency power spectrum. This makes a new trace with the square of the amplitude of each Fourier component. It is not clear why you would want to do this with a spectrum, but perhaps if you import snatches of a .wav file, peaks in the power spectrum would tell you what notes were being played.

9.6- Draw Fourier (frequency) axis. Generate Frequency axis (tic-marks and labels) for a transform. This is in case you want to look at the actual transform. It labels the X-axis with the frequency of the transform, starting at zero on the left up to half the sampling frequency on the right. The units are in cycles/nm, or whatever your base unit is. You can expand the scale to blow up a region, but this will redraw the axes so do that before making the frequency axis, or make the frequency axis again after expanding.

You will probably find that most of the amplitude is at frequencies below 0.1 nm^{-1}, corresponding to a period of about 10 nm. This is due to the bandwith of UV-vis absorptionpeaks

9.7. Inv FT with resampling. This evaluates the inverse FT at user-specified sampling interval over all or part of the spectrum. This is an alternative to the "Resample" in the operations menu. Since this version of scanedit cannot accommodate spectra of different sampling in memory at the same time, the result is written to a file in the ascii (.mat/.mtx) format.

Again the first box asks for the trace to be transformed. All frequencies will be included without damping, so if you want to truncate or damp you should damp the transform (9.3). This is probably a good idea to prevent aliasing, so will be included in tho operatio in the future.

The next dialog is like the "save ascii" dialog, with text boxes for starting and ending wavelength, sampling interval, and filename. The starting and ending wavelengths will be rounded off to interval multiples of the sampling interval. The wavelength range cannot extend outside the range of the original spectrum. When you dismiss this dialog the calculation is preformed. Nothing is plotted, but when it is finished an alert tells you the result has been saved, with filename and directory (the current directory unless you put a path in the filename).

9.8 Fourier Components. Same as inverse transform, but it stops after adding each component and plots the result so far, so you see the spectrum taking shape. If you check the box "Save Components" then it saves two series of spectra as it goes: `FCOMP.nnn` has all the cosine waves multiplied by the appropriate factor, and `FSUMM.nnn` contains the successive cumulative summations. If you select this option the number of components will be limited to 999, but you probably want to select much less than that to avoid 2000 files in one directory! The number of components used (and saved if you select that option) is approximately twice the number of points in the spectrum divided by shortest period you selected in the second box, as explained in the next section.

9.9 Details of the Fourier transform routines.

This is a discrete transform, meaning that it is calculated at discrete frequencies being multiples of 1 per period and discrete wavelength points. In order for that to work, the function being transformed needs to be periodic, and spectra are not. We can define the entire spectrum to be one period, but that

would require the last point to always be the same as the first, or result in a huge discontinuity between the last point and the first of the next period. Instead we define the period to be twice the length of the spectrum, and define the second half to be the first half, taken backwards. So the constructed period goes from first point of spectrum to last and then back to first, ensuring that it begins and ends at the same value.

Specifically 0 radians is taken as 1/2 interval before the first point in the spectrum and π as 1/2 interval after the last. From π to 2π, or equivalently from -π to zero, the spectrum is reversed: F(-x)= F(x). This means the constructed function is symmetrical about zero, a so-called "even" function. One consequence of this is that only cosine terms of the transform will have non-zero coefficients (Sin is odd: F(-x)= -F(x); so integrating a sin wave against our even function, -π to 0 will exactly cancel out what we get from 0 to π.

By the same token, integrating a cosine term, the value from -π to zero will be the same as from zero to π, so we only need to integrate from zero to π and double it. That being the case, we never actually construct the double spectrum, but integrate Cos waves over the actual range of the spectrum being transformed, and double the result. The first frequency is zero, and since Cos(0 × X) is always 1, we integrate against 1, or equivalently take the average of all points in the spectrum times the spectral range. Since all the remaining Cos terms are waves with average value zero, it is this zeroth term that determines the "DC offset" or baseline, of the spectrum. And since the spectrum consists of discrete points, "integration" is actually taking the dot product of the spectrum with a vector consisting of the Cos terms evaluated at the same points.

The next frequency is 1 cycle per 2π, i.e. 1/2 cycle within the spectrum. Then 2, 3, 4, etc cycles/2π . We store these coefficients in the next available trace, which has the same dimension as the trace containing the spectrum, starting with the zero'th element. So frequency zero (the constant offset) goes into point 0, frequency 1 in point 1, frequency 2 into point 2 etc.

Because we take zero as 1/2 interval before the first point, and π as 1/2 interval after the last point, the length 0 to pi is equal to the sampling interval × N, where N is the number of points. So by the time we get to the Nth order (N cycles per 2π), that is N/2 cycles per π. We just said π is equal to N intervals, so this frequency corresponds to 1 cycle per two samples, or half sampling frequency.

This is the Nyquist frequency, and frequencies at or above the Nyquist frequency cannot be recovered reliably from a sampled stream, so we stop with the highest order equal to N-1, where N is the number of points. Together with the zero-order term this means we have N amplitudes to store in the transform. It makes sense that a transform consisting of N amplitudes would be necessary and sufficient to store the information in a spectrum of N points, and it allows us to save the amplitudes in a regular trace like a spectrum. The transform can be saved to disk like a regular spectrum and reloaded at a later date to regenerate the spectrum with damping or resampling, but since the Fourier transform is so fast there's not much point in that.

It is conventional to think of Fourier terms in terms of phase and amplitude. For an even function the phase is always 0 or 180* (π), and the amplitude is positive. Here we take the phase as always zero, and the coefficient ("amplitude") can be positive or negative (changing the sign and shifting the phase by 180° gives the same curve). However if you are looking to get a frequency spectrum, you would want to take the absolute value, or for a power

spectrum the square, of all the amplitudes. The latter is what the "Power spectrum" option does.

For example, a spectrum from 250 to 650 nm with 10 points per nm has 400 nm so 4001 points. adding 1/2 interval on each end means π is 400.1 nm If you select the shortest period (wavelength) to be 10 nm, the highest frequency is 0.1 nm^{-1}, or multiplying by twice the spectral range, 80.02 per 2π, which will be rounded off to 80 cycles and the actual frequency will be 80/800.2 = 0.099975 nm^{-1}. There will be 81 terms with frequencies of 0 to 80 per 2π (0 to ~0.1 cycles/nm). The Nyquist frequency is half the sampling frequency, or 5/nm. This corresponds to 4001 cycles per 2π and will be the 4001'th term The transform stops one sort of that, with the 4000^{th} term. Together with the 0^{th} term, there are 4001 amplitudes to be stored.

Note the order has to be an integer, so period has to be an integral sub-multiple of the spectral range *after adding the two half intervals*, so npoints × interval. This means the frequency and period will not be round numbers unless the number of points is a multiple of 10. You might think that 500 - 600 is an even multiple of ten, but the number of points, inclusive, is 1001. Try deleting the last point (500-599.9).

Future plans (Fourier routines)-

put a dropdown menu for the Fourier operation on the console. Select a single trace then select the Fourier option and it will be preformed on that trace, rather than asking which trace.

Have a gui with a radio button to select option (already set if you chose the option in the menu), trace number to operate on (already set if you are coming from the console menu), and min period and shoulder frequency set to some default or blank)

Chapter 10. Smoothing and derivatives

These are both accomplished by fitting the spectrum to a polynomial in a sliding window, as described by A. Savitzky and M. J. E. Golay (1964), "Smoothing and differentiation of data by simplified least squares procedures," Anal. Chem. 36, 1627–1639. The sliding window is required to comprise an odd number of points, and the points are evenly spaced. For purpose of fitting, the central point is taken as zero. This means, given a polynomial like $a + bx + cx^2 + dx^3$, since x is zero at the central point, the value at the central point is just the coefficient a. The first derivative wrt x is b, the second derivative is 2c, and the third derivative is 6d. This works everywhere except within 1/2 the window width from either end, where we cannot make the points be the central value. There we evaluate the smoothed value and derivatives from the polynomial for the first or last window width in the spectrum, using all terms.

Since we are calculating the derivatives of the best-fit polynomial, in effect we get the derivatives of the smoothed spectrum rather than the original spectrum. This turns out to be a good thing, because it helps overcome one of the problems with taking higher-order derivatives of experimental curves: high-frequency components tend to blow up in the derivatives.

To understand this, consider expression of a spectrum in terms of its Fourier components. Just as the spectrum is the sum of its Fourier components, the Nth derivative is the sum of the Nth derivatives of the spectrum's Fourier components. The components are sine waves. The first derivative of sin(x) is Cos x, and the second, third, and fourth are -sin(x), -Cos(x), and sin(x) again. All of these have the same amplitude, so sin(x) does not blow up with successive differentiation. The derivative of sin(Fx), where F is the frequency, is Fcos(Fx), and the amplitude is multiplied by F with every successive differentiation. So for frequencies higher than 1 each derivative has higher amplitude than the last, but for frequencies lower than 1 each derivative is smaller than the previous. The period of sin(x) is 2π, So Fourier component with periods of < 2π nm (frequency greater than $1/2\pi$ nm^{-1}) will blow up with successive differentiation unless they are filtered out.

UV-vis spectra tend to be relatively smooth, peaks having width at half height about 10 nm or greater, so the components with period shorter than 2π result mainly from noise, and the actual shape of the spectrum is not likely to be affected by filtering out higher-frequency components.

Experiments with smoothing spectra consisting of sine waves suggests that with a cubic Pn, the amplitude is reduced to 98% when the window contains 0.5 cycles, and decreases rapidly at higher frequencies. With a 7th-order polynomial, that attenuation is reached when the window contains 1.5 cycles. The "cycles per window" frequency is just the cycles/nm times the width of the window, so if we want to preserve frequencies with periods of 5 nm (0.2 nm^{-1}) to within 98% (and lower frequencies better), then the window width should be about 1 nm for a cubic polynomial and 3 nm for a 7th order Pn. It is also important that the number of points in the window should be significantly greater than the order of the polynomial, or avoid getting exact but nonsensical fits. These windows should be alright if the sampling frequency is 10/nm (11 points for the cubic, 31 points for the 7th order). However with coarser sampling it may be necessary to use wider windows. In the end this should just be a starting point - try several different values around these and you should be able to judge when the

spectrum is being smoothed and when it is being distorted. A test is to subtract the original spectrum from the smoothed one, and see if the difference looks like noise or something systematic.

For cytochrome spectra, whose peaks are significantly wider than 10 nm at half-height, We have good results with 1.4 to 2 nm windows using a cubic polynomial.

There are two menu items: "differentiate" and "more derivatives". The first is using an equation for the best fit cubic polynomial (only valid for evenly spaced points centered on zero). Thus it yields the smoothed spectrum and the first three derivatives. The second numerically evaluates each term at each point, and finds the least-squares best fit to a sum of x raised to various powers, so can determine more than three derivatives.

*This might be predicted from the power series: $\sin(z) = z - z^3/3! + z^5/5! - z^7/7! \ldots$ For z<=1 this should converge rapidly due to the factorial in the denominator (5! is 120, less than 1% of the first term for x=1). To take into account frequency, for the power series of sin(Fx), where F is frequency, we have z= Fx. But that frequency F is in cycles/2π, since F=1 gives a period of 2π. In terms of cycles/nm the frequency is 1/2π nm = f = F/2π, or F =2πf or 2π/p

F = 2πf = 2π/p, where f is nm^{-1}, p is in nm.

sin(Fx) = sin (2πf x) = sin(2π/p × x)

So we are asking when 2πx/p = 1 and it is when x = p/2π or p= 2πx

Since we evaluate the Pn in a window around zero, the max value of x is half the window width, so 0.7 for a window of 1.4 nm. This would give 2πx/p = 1 when p = 4.4 nm

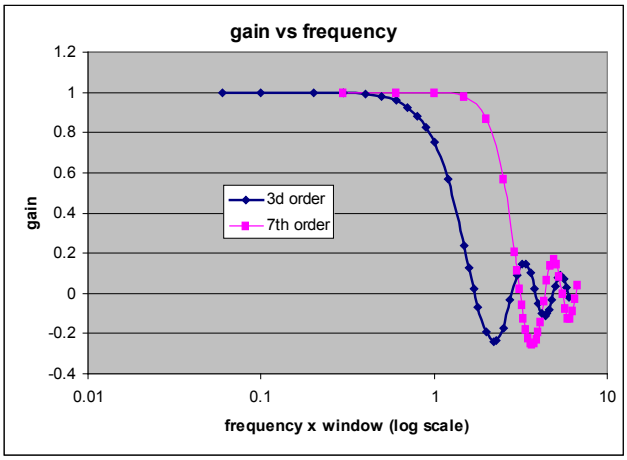
Our empirical observation of 0.5 and 1.5 cycles per window, given that 0 is in the center of the window, give 0.25 and 0.75 cycles for max x.

say the wavelength is 10 nm as in the experiment. 0.25 cycles is 2.5 nm = x

Fx = 2πx/p where x/p = .25, Fx = 1.57 x/p = .75 gives Fx = 4.7

But two things may make the problem less serious than expected-

1. the max x is only encountered at the wings of the window, and the fit may be pretty good at the center where the smoothed value is taken.
2. The polynomial terms are not orthogonal over the window. Lower order terms may partially compensate for missing high orders, improving the "smooth" function but not the derivatives.



Critical frequency for Taylor to converge-

Period of wave =2pi x windowwidth is a frequency of 1/(2Pi) per window or 0.159 cycles/window. At this frequency the filter has near unity gain, but somewhat higher frequencies are nearly unity gain. "gain" just means amplitude of the smoothed spectrum relative to original. Filtering a sin wave with varying window width.

Chapter 11. Scripting, to automate tasks in ScanEdit

Prepare a text file with commands from the below list, then select "run script" under the "disk-based" menu. A file selection box opens. It is set to look only for files with extension .scr, but you can change the filter to look for text files or for "all files" Select your script.

Commands and syntax:

(command and parameters should be separated by commas and/or newline

CD, [D:]path (if D: included, first chdriv D:)

CD, dir chdir to the dir. It is likely that process of selecting script file has moved curdir to that dir, but not changed datadir. Should this change datadir as well? Or always have script in the dir with the data?

LSINV, least-squares inverse of non square matrix (rows > col I think) => left inverse

MATINV, - invert a square matrix

MATMULT, (spectra x mat, maybe should be specmult)

TRANSPOSE, file1.mat, out.mat. (out.mat can be the same as file1.mat)

SUBMAT, filein, fileout, row1, row2, col1, col2

(use "0" for default use first (row1 or col1) or last (row2 or col2). 0,0,3,0 just omits 1st 2 columns. Multiple sets of arguments can be given after the command, (MUST BE term by blank line!).

DIAG, file1.mat, out.mat 'reads a column or row vector and puts out diagonal matrix.

(out.mat can be same as file1.mat to overwrite) (outer product of V with [I]?)

MATMULT, file1.mat, file2.mat, out.mat (file1*file2, file2 on the right)

MAT2CSV, file.mat, file.csv

DIAG, (onalize) Filein, Fileout - reads a vector and makes square diagonal matrix with those elements

EXIT, - leave without reading the rest of the file (optional). Allows explanatory comments below EXIT,

-operating on spectra in memory:

SHIFT, (tracenumber, first shift, last shift)(shift,1, -3, +3) This creates a series of shifted traces. The shifted traces are not kept in memory but written out to series "shifted.###".

MULT, trace#, multiplier (used for dilution correction)

DIV, trace#, divisor (used for dilution correction)

REPLIC, trace1, N create N copies of trace trace1. used in preparing shift series. NOT used in shift series above!

LINCOMB: make spectra as LC of spectra in memory, writes to disk (not yet implemented)

Some routines will read multiple lines of param without repeating the command. Include a blank line (or end of file) to terminate.

Others only read one set of parameters and exit. You don't need to leave a blank line (next line can be another command), but it won't hurt.

For either kind, the command can be on the line with the (first) arguments, followed by comma, or on a separate line before the arguments, not followed by comma (newline separates). If there are multiple arguments, they should be separated by commas (or newlines)

(Items are read by input# one at a time. with input#, strings are terminated by comma, CR, or LF. So if items separated by space or tab, the whole line is read as the command, not recognized, and ignored. On the other hand numeric input is terminated on spaces, so comma not necessary (but won't hurt). After comma, CR, LF the next item starts with the next non-blank character. except I think CRLF is read as a blank string?)

For commands operating on files (matrix or vector), the format for those files is the old ".mat" format from which the .mtx format was derived for matrices, it is the same as

.mtx but without the parameters at the end. First 3 lines have 2, n, m where n is the number of rows and m the number of columns. This header is followed by elements of each row, one row at a time (i.e. column fast, row slow). For a vector, 2, 1, m (row vector), or 2, n, 1 (column vector.

Appendix A. - Setup, Installation

This document is divided into two sections. The first (very short) explains what you may need to do to get the scanedit program to run on your computer. The second involves things you can do, if you want to use it routinely, to make its use more convenient. Please report any difficulties you encounter.

1. Getting scanedit.

Download ScanEdit from https://sourceforge.net/projects/scanedit/files/openspectralworks-1.0/. Click on "Download latest release" to download the file "tutorials.zip". Or you can get it directly with this link: https://sourceforge.net/projects/scanedit/files/openspectralworks-1.0/tutorials.zip

Make a directory somewhere and unzip the files to that. This file includes, in addition to the tutorials, the executable (ScanEdit.exe) and a runtime library (comdlg32.ocx) that will be needed unless it has already been installed with some other software.

For testing, you may want to make as few changes to your system as necessary. With that in mind we suggest three successively more invasive options:

1)Just run it. (probably won't work on modern computers)

2) run install.bat to just install the .ocx dependency. (should be sufficient)

3) run setup.exe for full install (which can be uninstalled via Windows add/remove_programs).

On some older machines, no setup will be required. You can open a folder or memory stick on which you have downloaded the "Tutorials" package, find the executable (scanedit.exe) in the top folder, double click its icon, and it will run. If this is the case you can run the tutorials as described with no further setup.

On other machines the program will start, displaying the graphics screen, but there will be a message box warning that the program cannot run because comdlg32.ocx is not present or not properly registered. In this case you need to open the **setup** folder (also in the tutorial package). Right-click on **INSTAL32.BAT** or **INSTAL64.bat** (depending whether your operating system is 32-bit or 64-bit) and select “run as administrator” . Hit <enter> several times when prompted to do so. A box should pop up saying “comdlg32 has been successfully registered”. Note on Win7 and later this has to be run as administrator, not just a user with administrative privileges. Hence the instruction to right-click and “run as administrator”. Do this from a local drive (locally mounted USB stick is fine) because your network drive may not be mounted for administrator,

That will suffice in all cases I have seen so far. However if that fails, open the “full install” folder and run the setup.exe program found there. This is a regular install program like Installshield or Wise. It will install the executable in its own directory in “program files (86)” and give you a shortcut in the start menu. It will install and register comdlg32.ocx and if necessary msvbvm60.dll. and it will install or update if necessary six other windows system files that are involved only in getting the setup program to run.

Explanation: scanedit needs only two other files in order to run: the runtime modules (msvbvm60.dll (ms visual basic 6 virtual machine) and comdlg32.ocx

(common dialog activeX controls). Msvbvm60 is a standard component of windows vista through Win11, so you should already have it. Comdlg32.ocx is included with the VB6 development kit and is meant to be distributed and installed together with any software that uses it. However quite a lot of software does use it, so machines with a lot of software installed are likely to already have it. Newer software does not use it (.com being replaced by .net). So newer machines may need it. However I checked on the Win7 machines in one of the computer labs here at Upstate Medical University, they have the comdlg32 and scanedit runs with no setup required (but that was 2012 and Windows 7).

Here is what gets installed or updated (if current version is not present) if you need to run the full install program **setup.exe** (a complete list is in the SETUP.LST file in the same directory as the setup program):

1. bootstrap files required to run the setup program
2. program to setup the bootstrap and start the setup program
3. setup program itself
4. scanedit.exe – the executable
5. files required to run scanedit
   - msvbvm60.dll – the runtime module (if not present)
   - comdlg32.ocx – The common dialog functions (file open/save box)

Some of these files probably already exist. The setup program will not overwrite a newer version with an older one without asking for confirmation first (say no). If it does need to replace an older version and that file is in use, it will need to reboot in order to do so. Otherwise installation does not require a reboot.

2. Update to latest "nightly build". The above was to set up the "latest release", scanedit-1.0. However that was from 2018, and a lot of development has occurred since then. Many functions described in this manual did not exist or were implemented differently in that version. It is high time for another minor release, and the current version is basically release Candidate for -1.1. as soon as I get around to updating the rest of the package.

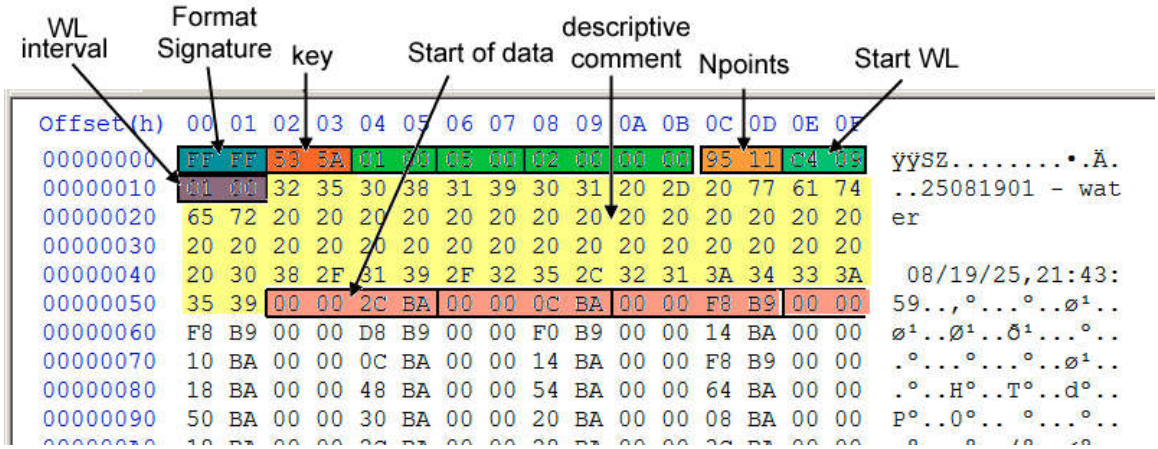
To get the latest version, go to:
https://sourceforge.net/projects/scanedit/files/NightlyBuilds/ and download the executable ScanEdit.zip. Unzip it to replace the old version of scacnedit.exe. You can also download the latest version of this manual, Scanman.zip, which unzips to a pdf.
For direct links to the files:
https://sourceforge.net/projects/scanedit/files/NightlyBuild/ScanEdit.zip
https://sourceforge.net/projects/scanedit/files/NightlyBuild/ScanMan.zip

3. Making scanedit convenient -

Suggestions for placing a shortcut on the desktop, adding to start menu, associating the program with its files, and setting various preferences are discussed in Chapter 5

Appendix B. - Description of the Format of scanedit native files

1. Binary files- The absorbance data is stored as 4-byte floating point numbers (IEEE 754 32-bit). This makes it more compact than the .mtx files which stores them as formatted decimal strings. Each file stores a single spectrum. The file starts with 9 2-byte integers. The first is hex FFFF (-1) signature for the format. The next five integers are intended to store metadata bout the spctrum, The first of these gives some clue about the origin of the spectrum- key is below. The next 4 are available to store other parameters. In the case of the files converted from Shimadzu .spc files, they store the scan speed, bandwidth, and sampling interval. Otherwise they have not been widely implemented.



After those 6, there are three critical 2-byte integers, holding the number of data points in the spectrum, the starting wavelength, and information about the sampling interval. The starting wavelength is always the short-wavelength end, and is expressed in internal WL units. The internal unit is the reciprocal of the variable "ppnm" (points per nm). "ppnm" is normally 10, in which case the stored starting wavelength is an integer equal to 10 times the actual starting wavelength. Since it is stored in a 2-byte integer, this means the starting wavelength cannot be larger than 3276.8. This will change in the next version of the format.

The final integer defines the sampling interval in terms of the internal wavelength unit. With ppnm = 10, 1 implies 0.1, 2 implies 0.2, etc. Since it is an integer, the sampling interval has to be a multiple of the internal wavelength unit. if ppnm is 10, you cannot have 4 or 8 points per nm, or 20. If you load a .mtx (ascii) file with such sampling interval, the program will automatically change ppnm to 8, or 20 and all will be well. The current binary format does not save ppnm (this too, will change . . .), but the program allows you to set ppnm if you know the correct sampling interval. This is done with the "redimension" option (3.1.3.1.8) under the File menu. At the bottom of its dialog is a textbox labeled "MaxPPNM" which will initially read 10. This is actually the ppnm variable, and if you change this (without changing the value for pmax above), no redimensioning will occur and the spectra will not be erased from memory. Clear the screen to display the correct wavelength scale, and replot your spectra.

After these 9 integers, 62 bytes are used to hold the descriptive comment associated with the spectrum. After that the data begins, 4-byte IEEE 754 floating point numbers.

Assigned values for the second integer (FPARAM(j,2)). These may change:

0 anything where we haven't yet assigned
5 signature for calculated spectrum (lin combo) including ave, sum
6 signature for calculated spectrum (Integral)
8 signature for residual spectrum
20 svd singular spectrum
21 svd reconstructed spectrum
22 svd real spectra from fit of singular vectors
23 signature for spectrum calculated by matrix multiplication
30 signature for Fourier transform
23 signature for generated spectrum
23123 (0x3A33): Spectrum from Shimadzu

2. .mtx ascii files – The first three lines contain a single number each, respectively the number two, the number of spectra, and the number of points in each spectrum (each spectrum has to have the same wavelength sampling).

These are followed by the absorbance values, first all the points from the first spectrum, then all the points from the second spectrum, and so on. The values can be separated by spaces, commas, or newlines (i.e. it is alright to have a single value on each line).

After all absorbance values of all spectra, then comes a decimal number representing the wavelength of the first point of the spectra, then a second decimal number representing the wavelength increments between points. Finally, one line for each of the spectra, containing the descriptive comment.

For example if there are three spectra running from 340 nm to 650 nm with 10 points per nm, the file would look like:

```
2
3
3101
(absorbance data - first point of spectrum 1)
.
.
.
(last point of spectrum 3)
340
0.1
descriptive comment for spectrum 1
descriptive comment for spectrum 2
descriptive comment for spectrum 3
```

Editing the .mtx spectrum file is a convenient way to shift the spectra to make up for wavelength miscalibration. If you decide, from the way the fits are going, that the dataset was taken on a spectrophotometer reading too long in wavelength by 0.1 nm (what was read at 340 nm was really 339.9, you can make a new copy of the .mtx file and edit it to make the starting wavelength, for example, 340.1 instead of 340. This will cause the first point of your standard spectra, which is really 340 to be matched with 340.1 in the spectra being analyzed, which is really 340. However your last point will now be at 650.1, and if

the spectra being fit are nominally from 340 to 650, you will have to truncate at 650. Just load the modified .mtx and resave it as .mtx, specifying 340.1 to 650 nm.