EXAFS Analysis Organization and Data Flow

Matthew Marcus

May 2, 2002

This writeup shows how the EXAFS and other data analysis programs are organized and the sequences in which they can be used. There are five main groups of programs, MCA, XRF, EXAFS front-end, EXAFS Fourier and general-purpose 2-column data manipulation. Two-column files are used for data which is expressed as a quantity plotted against some other quantity and consist of two columns of ASCII numbers with no header. Such files are generated by most of the programs.

The functions of the five groups are:

MCA: Reads and manipulates MCA data from the detector

XRF: XRF elemental mapping and the analysis thereof

EXAFS front-end:
EXAFS data from raw scans up through unfiltered 'wiggly-part' data.

EXAFS Fourier:
Fourier transformation and handling of resulting files including non-linear least-squares fitting.

General purpose e-column data manipulation:
Editing of XY data, plotting, weighted sums or products

Files are kept track of through the mechanism of file extensions. A list is given at the end of this note. Most programs are set up so that their file-selection dialogs default to an extension appropriate to the type of file. Thus, the k-space&background-remove program produces a `.b` file by default, and the FT program takes `.b`'s as input.

The first set of programs I'll consider is the MCA set. This consists of the MCA utility on the data-taking machine and the MCA reader on the analysis computer. The MCA reader lets you see the spectrum from one or a sum of detectors and puts out a 2-column file for the spectrum. This simple dataflow is shown in Figure 1.

Data Flows: MCA Programs

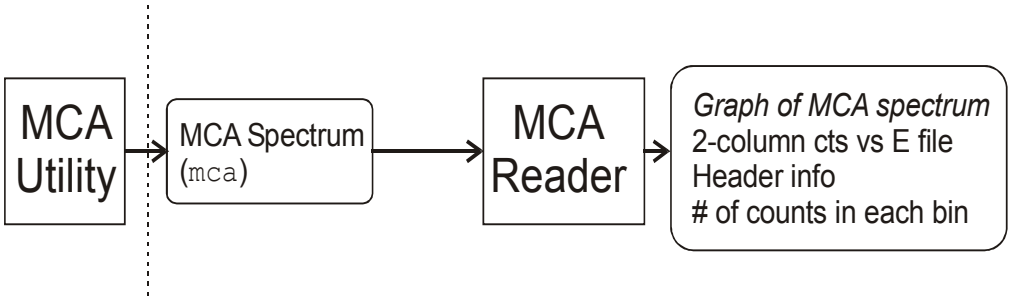

Figure 1. Dataflow for the MCA programs. The various types of objects shown are as follows and are the same for subsequent Figures:

Rectangular box:	*Program*
Dashed line:	*Separates data-taking from data analysis programs*
Rounded box:	*Results. Within this box, we have:*
Text	*Results or files*
Italic text	*Most important files or results*
(`ext`)	*File extensions*

The dataflow for the XRF programs (Figure 2) is not quite so simple. The fluorescence-mapping program on the data-taking machine produces a file which contains the maps for all SCA's. It's essentially a three-dimensional array (SCA, x,y) with a header. This can be looked at with the display program. The display program is the main means of interacting with these files. However, you can also do difference maps with another program. These maps have the same format as the normal map files, so they can go into the XY display program. Further, there is some processing, such as making cross-correlation maps, which can be done with the display program. The results of such processing again look enough like raw data to be readable by the same program, which is why the Figure shows a loop. The end results will be images of maps in single elements, tricolor maps in three elements, scatterplots or correlation images.

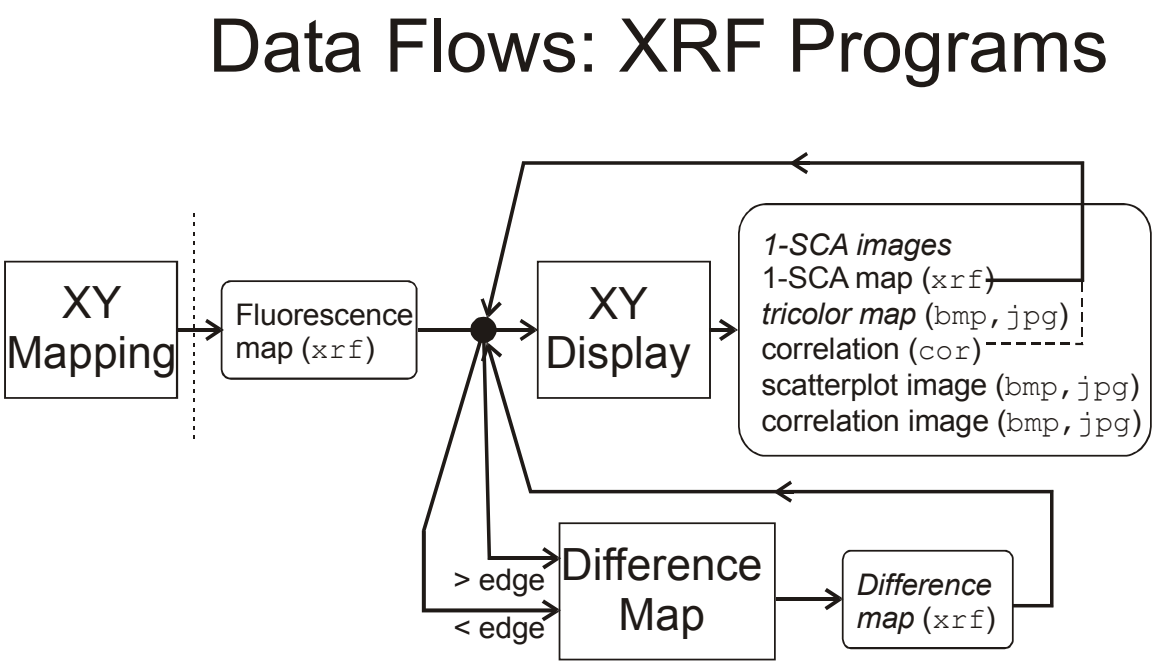

Figure 2. Dataflow for the XRF program. Note that the Difference Map program takes two `.xrf` *files as input. The dashed line coming from the* `.cor` *correlation file is meant to indicate that this is a minor path. Not shown: Ratio map, which works just like Difference map.*

Next, we have the front end of the EXAFS process, as shown in Figure 3. This is considerably more complex. There are two main types of files, full-information (`dat`) and 2-column. The full-information files have all the information from the beamline, including counts and count times for all counters. This information is needed for operations such as summing and deadtime correction, but is removed when going to the next stage of analysis. Thus, if the spectrum was acquired in transmission mode, the full-information file will have both the I_0 and I_t counters, but the rest of the data-analysis programs require only $\ln(I_0/I_t)$. Thus, the principal output of the EXAFS Data Editor is a two-column file with the energy and $\ln(I_0/I_t)$ as the columns. These two-column files are denoted by a `.r` extension.

The raw signal file then has to be converted to an EXAFS (k vs. $k^n\chi(k)$) 2-column file. This involves subtracting the pre-edge, choosing an E_0 value, running a spline through the post-edge, and subtracting and dividing by that spline. All these steps are done using the k-space&background removal program. The resulting files have a `.b` extension. This program can also be used to produce files (`.t`) suitable for XANES analysis.

Data Flows: EXAFS Front End

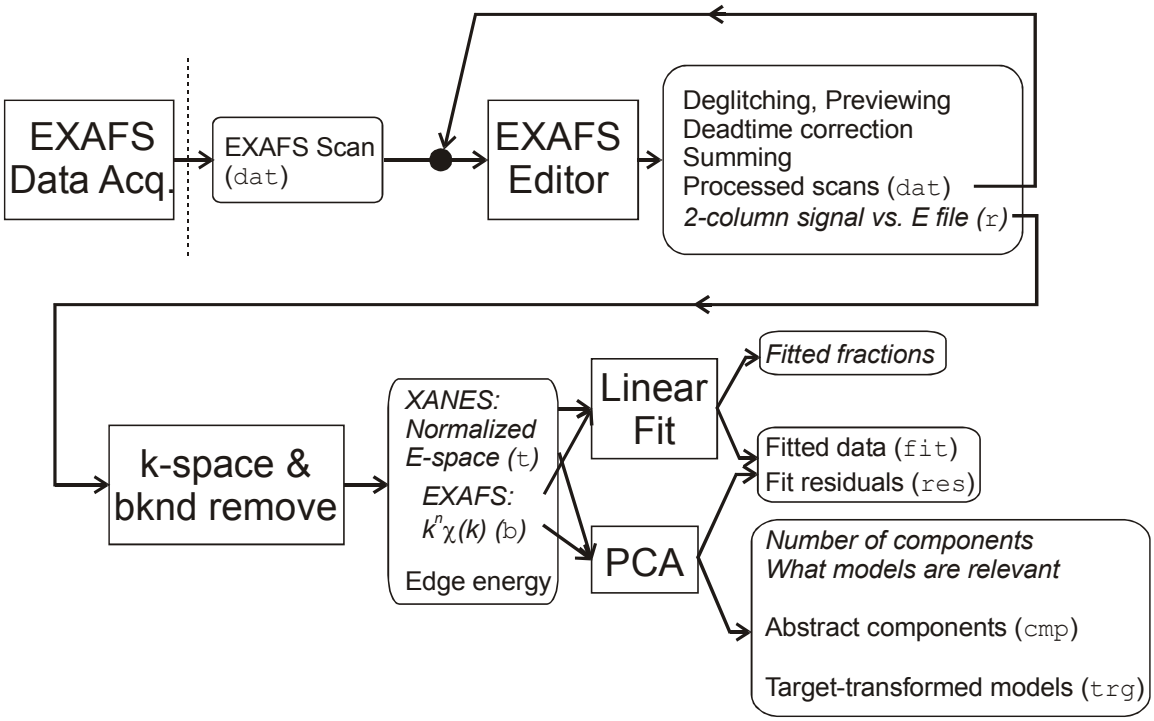

Figure 3. Dataflow for the first part of EXAFS/XANES analysis. This Figure covers steps up to but not including Fourier manipulations.

The .b and .t files can be used for 'fingerprint' or 'physics-free' analysis by linear fitting or PCA. All fitting programs, including PCA, produce best-fit (`fit`) and residuals (`res`) files. In addition, the PCA program produces files for the abstract components (`cmp`) and any target-transformed models which the user may have tested (`trg`). All these files are two-column, with *E* or *k* as the abscissa.

Analysis can also proceed along the Fourier track (Figure 4) in which one or more shells are extracted by Fourier filtering then fitted to amplitudes and phases derived from previously-extracted shells. The FT program can save out filter-function files so they can be re-used, thus assuring consistent filtering between different samples or between samples and models. Also, FT can save the filter residual, which is the input signal minus the filtered: `2b` = `b-f`. This residual can be put back into the FT and re-filtered to see higher shells. The Multishell fit program uses filtered or unfiltered data (usually filtered) and model amplitude and phase files, and produces the coordination numbers, distances, etc. These parameters can be saved into a file (`fpr`), which looks like a configuration file, and can be read back into the fit program. This program also produces the usual `fit` and `res` files.

Data Flows: EXAFS Fourier

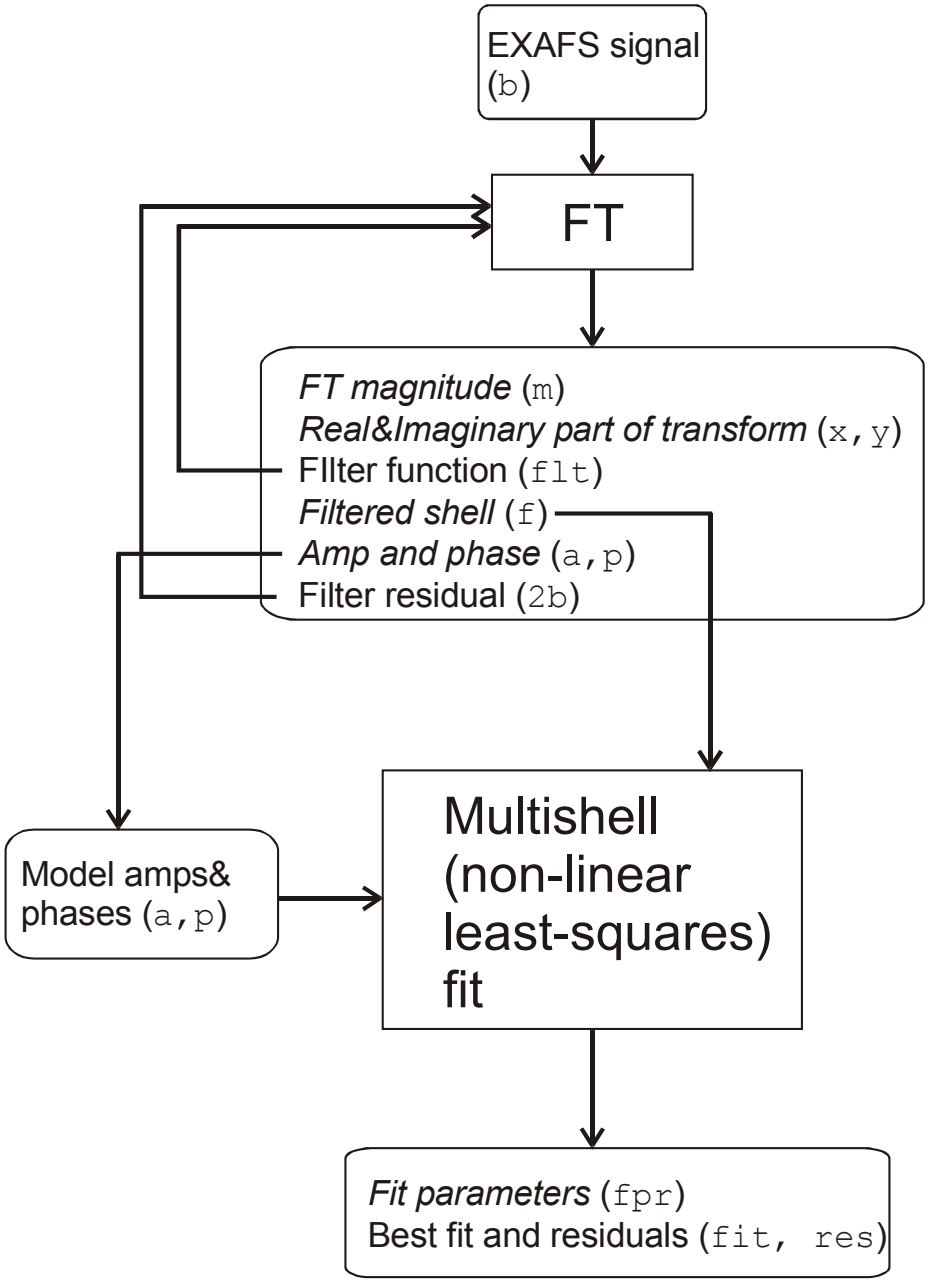

Figure 4. Dataflow for Fourier (shell) analysis. The FT *program can re-use the filter residuals (*`2b`*) as input and the filter functions (*`flt`*) for filtering another file. The fitting program uses amplitudes and phases which usually come from EXAFS (*`b`*) files other than the one being analyzed.*

The final group of programs consists of those which deal with two-column files in a general, non-specialized way. Presently, this group consists of the plot-add-multiply

and 2-column editor programs. There is no Figure for these because there is no set path by which they are used.

The plot-add-multiply program has two main functions. One of these is to plot a set of files on a common scale. This is useful for comparing files, plotting the real and imaginary parts of the FT along with the magnitude, etc. The full set of graph controls is available and the graph can be saved as a bitmap. The second function is that it does weighted sums or products. Each file is viewed as a tabulation of a function $y=y(x)$, and the program lets you make $Y=\Sigma w_i y_i(x)$ or $Y=\Pi y_i(x)^p i$, interpolating and pruning as necessary to put all the values on a common grid. This is handy for tasks such as adjusting phase shifts for semi-empirical modeling and making linear combinations of models to simulate spectra.

The 2-column editor works on one file at a time, but has more functions than plot-add-multiply. It lets you cut a piece out of the middle or cut off the ends of a set of data, resample by interpolation, sort, and do arithmetic on the ordinate or abscissa. The arithmetic operations supported are: add a constant, multiply by a constant, take the log, take the exponential, get the coordinates of a cursor, and use cursor coordinates in any of the above operations. The program also has functions for integration, differentiation and removal of a straight-line background.

Now I will describe some examples of how these programs can be used together for data analysis tasks. First, consider a signal-vs-E spectrum with a strongly curved post-edge background, and suppose you want to normalize it for XANES applications. A quick-and-dirty solution can often be done with 2-column editor. First, subtract off the pre-edge background. Then, take the log of the data. Fit a linear background through the part of the post-edge past the XANES region. Now remove that background and take the exponential. This set of steps is equivalent to dividing by an exponential curve. The fit to the curved post-edge is usually good enough for XANES work.

For a next example, suppose you need a model for a Cu central atom surrounded by Ti scatterers (Cu->Ti) in a metallic environment. There's no suitable intermetallic in this system. You might think to use FEFF to get ab-initio values, but that doesn't take into account the various distortions and artifacts of data reduction. A better way is to run FEFF twice, once for Cu in Cu metal (Cu->Cu) and once with a hypothetical structure consisting of Cu surrounded by Ti. You can generate the FEFF input file for this by starting with a file made by ATOMS for Ti metal and editing it to change the central atom to Cu. Now, create the phase and amplitude files for these two theoretical models, in the first shell. Call them `cucuth.[ap]` and `cutith.[ap]`, where the `[ap]` means either `a` or `p`. Also, get phase and amplitude files for Cu foil (cufoil.[ap]). Now do the following manipulations with plot-add-multiply:

```
cuti_semi.a = cufoil.a*(cutith.a/cucuth.a)
cuti_semi.p = cufoil.p+cutith.p-cucuth.p
```

The product and ratio can be done as a weighted product with powers of 1,1 and –1. The resulting files will be appropriate for Cu looking at Ti. at the distance and coordination number specified when generating `cutith.[ap]`. The E_0 will be approximately correct and all Fourier and spline-fit artifacts will be taken into account. These files are ready to be used in Multishell fit. Of course, this method must be used with eyes open, since it assumes things about transferability. For instance, you wouldn't

use it to go from Cu->O in copper acetate to Fe->Mo in an intermetallic because the chemistry of the site is very different and the atomic numbers far apart.

The Stern ratio method for evaluating Debye-Waller effects is another example. In this method, you plot the ratio of two amplitudes vs. k^2 on semilog axes. How do you get the abscissa to be the square? First, use plot-add-multiply to take the ratio ($A_1^1 * A_2^{-1}$). Next, take this ratio file into 2-column editor and do the following sequence: x->ln(x),x->$2x$,x->exp(x). This turns x into x^2. Now, bring it back into plot-add-multiply and plot semilog or do y->ln(y) in plot-add-multiply and plot on a linear scale. I don't have a linear fit routine which returns the fit parameters, but you can use any data-graphing program for that, even Excel.

As a final example, consider the case of a sample with a weak second shell. This is common in natural materials in which the disorder is significant. When you take the FT, you often find a big first shell, and some hint of a second shell, but it's not really well-resolved from the first shell. One way to deal with this is as follows:

1. Use FT to get the first shell and save the filter residuals. This contains the second-shell information.
2. Look at the filter residual file with 2-column editor. You will probably find that the recognizable signal extends up to a lower k than the first-shell signal does. Chop the file so that the noisiest parts are gone.
2. Put this back into FT and see if there's a recognizable second shell peeping out.

Alternatively, if you have a model for the first shell, use Multishell fit to fit the first shell, then use plot-add-multiply to subtract the fitted first shell from the unfiltered data. Proceed as in steps 2 and 3 above. This procedure makes sure that you're only subtracting off something that really belongs to the first shell. This method can be used iteratively to split poorly-resolved shells such as the first and second of Fe metal.

To wrap up, I'll go through a highly-condensed example of what a typical set of data acquisition and analysis might be like. Consider a soil sample which contains a metal, say Zn, in some unknown form or set of forms. You want to know how many different forms there might be and what they might be. First, use the MCA Utility to get an idea of what elements are present. Drive around the sample a bit. If you find an interesting spectrum, save it out so it can be read later. Next, map the sample using XY Mapping. You will probably find interesting patterns of correlation, for instance that Zn is found where Mn or Fe are. Use this map to pick out spots for EXAFS. Tiny hotspots may be small grains of minerals which represent only a small fraction of the whole, so you may want to avoid being tempted into looking only at the strongest spots.

Now, take EXAFS data on each of several spots. The scans will be named things like `spot1001.dat, spot1002.dat`... and so forth. Use the EXAFS Data Editor to clean up the files, discard the bad ones, do deadtime correction, and add up the good ones. Do the deadtime correction before adding up. Now, you have summed scans for each spot, `spot1.dat, spot2.dat`...etc. Apply k-space&background remove to put it into k-space according to your favorite prescription for E_0 selection and background removal. Do it consistently. Consistency of methods is key in EXAFS analysis,

especially if you're doing PCA. You can now compare visually with references or use PCA on `spot1.b, spot2.b`, etc. to see how many independent components there might be. Use the target transformation to test references for plausibility. Alternatively, you can use the linear least-squares program to do fits and see what species are plausible. If you find a species that doesn't match your database, then you can dig deeper using Fourier methods and your deep knowledge of Zn minerology (sorry, software can't do everything!) to figure out what's there.

One last hint: Make up a readme.txt file as you go during analysis so you can remember what you did, what you did it to, and why you did it. It's a lifesaver when you go back months or years later.

Finally, here's a list of file extensions:

`mca`	MCA spectrum
`xrf`	XRF element map
`cor`	Cross-correlation map from XRF
`bmp,jpg`	Bitmaps from programs which produce graphics
`dat`	Raw, multicolumn EXAFS
`fpr`	Fit parameters from Multishell fit

------- Files past this point are all 2-column and may be input to any 2-column program---

`r`	2-column signal-vs. E EXAFS (raw, not background subtracted)
`t`	Background-subtracted, E_0-shifted E-space
`b`	EXAFS ($k^n\chi(k)$ vs. k)
`fit`	Fitted function from any fit program
`res`	Residual from any fit program, (input)-.fit
`cmp`	PCA abstract component
`trg`	PCA target transformed reference
`m`	FT magnitude
`x`	FT real part
`y`	FT imaginary part: $m = \mathrm{sqrt}(x^2+y^2)$
`flt`	Filter function (R abscissa)
`f`	Filtered, back-transform: $f = FT^{-1}\{flt(r)*(x(r)+iy(r)\}$
`a`	Back-transform amplitude
`p`	Back-transform phase: $f(k) = a(k)\sin(p(k))$
`2b`	Filter residual: `2b = b-f`