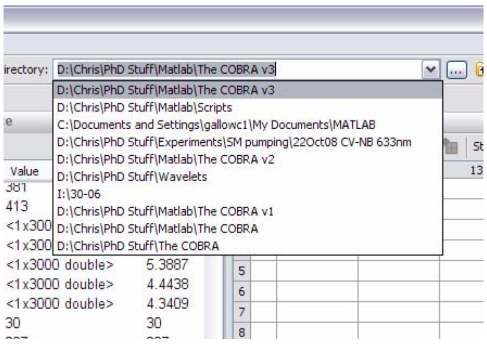
The COBRA: User's Manual

For any questions/comments/suggestions contact:
chris.gallow@gmail.com or Pablo.etchegoin@vuw.ac.nz

1. Opening the Program

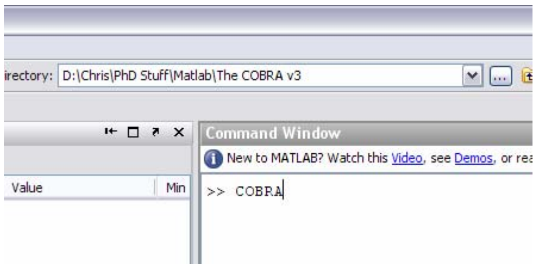
- Set MATLAB to use the directory in which the COBRA files are contained:



```
irectory: D:\Chris\PhD Stuff\Matlab\The COBRA v3
D:\Chris\PhD Stuff\Matlab\The COBRA v3
D:\Chris\PhD Stuff\Matlab\Scripts
C:\Documents and Settings\gallowc1\My Documents\MATLAB
D:\Chris\PhD Stuff\Experiments\SM pumping\22Oct08 CV-NB 633nm
D:\Chris\PhD Stuff\Matlab\The COBRA v2
D:\Chris\PhD Stuff\Wavelets
I:\30-06
D:\Chris\PhD Stuff\Matlab\The COBRA v1
D:\Chris\PhD Stuff\Matlab\The COBRA
D:\Chris\PhD Stuff\The COBRA
```

Figure 1

- Type "COBRA" in the MATLAB command line:



```
irectory: D:\Chris\PhD Stuff\Matlab\The COBRA v3
Command Window
New to MATLAB? Watch this Video, see Demos, or rea
>> COBRA
```

Figure 2

2. Importing data

- The User Interface should look like so:

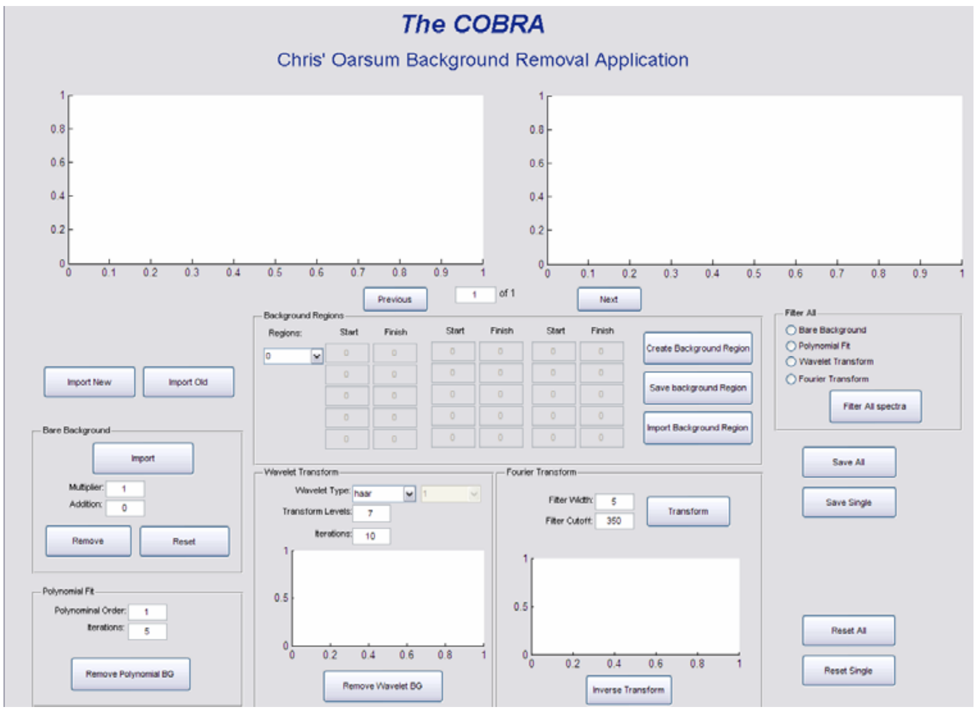


Figure 3. The complete "COBRA" user interface

- To import a new set of data click "Import New" and select the appropriate data file. The data must be contained in a .txt file in which the first column is the x-axis units and each subsequent column is a new set of data which have the same x-axis scale.
- After the data is imported, a plot will appear on the left set of axes showing the signal for the largest axes. The total and current event numbers will also be displayed below the plots:

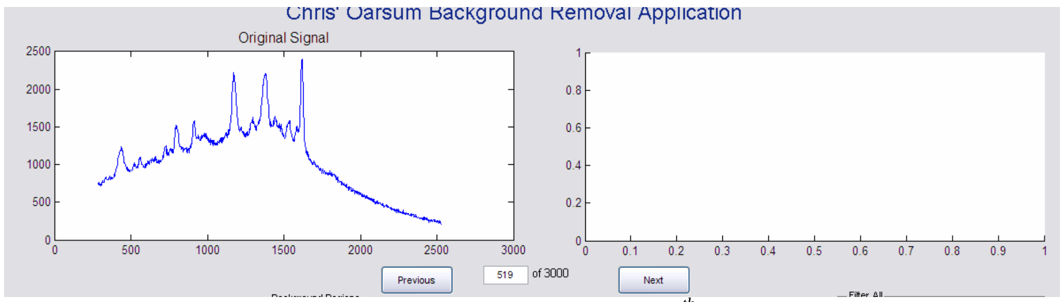


Figure 4: The largest signal of the imported data. This is the 519th event of a total of 3000 events as is shown below the axes.

- If the user would like to perform the fitting on a different signal event then they may browse through the signals using the previous and next buttons or enter a new event in the text box in figure 4.

3. Importing a Background

- If there is a file that contains a purely background event which you wish to remove from all of the signal events then this can be imported using the "Import" button in the "Bare Background" frame:

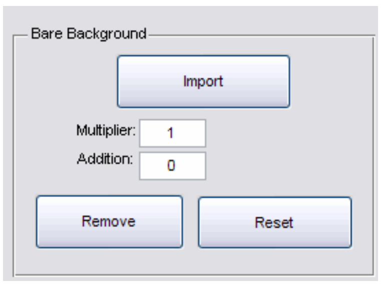


Figure 5. The bare background frame

- The background file must be a two column .txt file where the first column has the same x scale as the original data and the second column is the background signal.
- After the file has been imported, the signal event with the background subtracted will appear on the right set of axes:

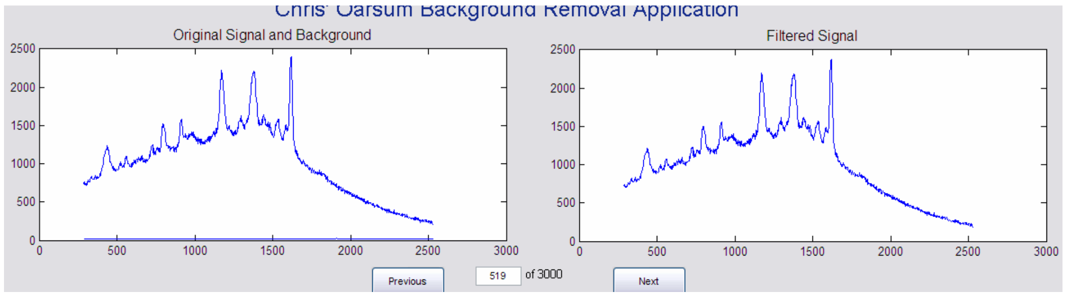

Chris' Oarsum Background Removal Application
Original Signal and Background
Filtered Signal
Previous
519
of 3000
Next

Figure 6. The original signal event with the background removed. Note that the background in this case very small relative to the signal.

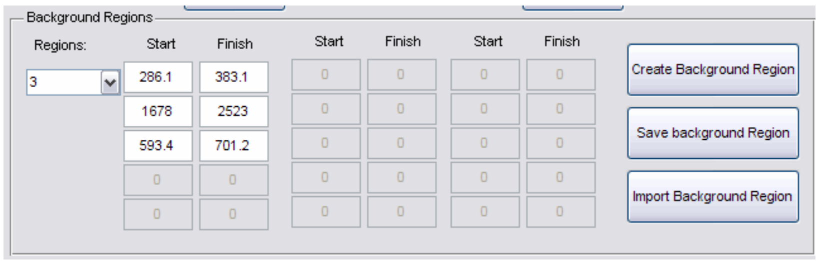
- The background may be modified by changing the multiplier and addition text boxes. The new background will be of the form

New BG = Multiplier*Original BG + Addition

- After the multiplier and addition values are set, the new background can be subtracted from the original signal by clicking the “Remove” button. This can be performed as many times as necessary.
- Pressing “Reset” will restore the signal to where it was without any background subtracted.

4. Defining the Background Region

- To assist with the polynomial and wavelet transform fits, it is often useful to define regions on the signal event that are completely background. To do this select the number of regions you wish to define from the list box and type in the start and finish x-coordinates for each region:



Background Regions

Regions: 3

Start	Finish	Start	Finish	Start	Finish
286.1	383.1	0	0	0	0
1678	2523	0	0	0	0
593.4	701.2	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 7. Background regions frame with 3 background regions defined

- Clicking on the data points on the plots will give the x and y coordinate at that point to help with estimating the background regions:

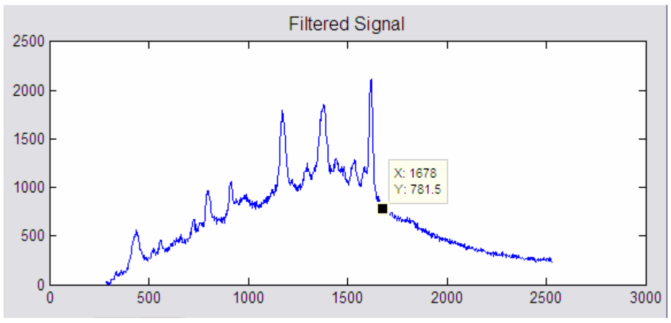


Figure 8. The filtered signal after a point has been selected.

- Once you are satisfied with the background regions, click “Create Background Region”. The background region can be modified just by changing the values and number of regions and clicking “Create Background Region” again. Clicking “Reset Single” or “Reset All” will not reset the background region. It is also possible to save the background region in a MATLAB file for future use and import it again using the “Save Background Region” and “Import Background Region” respectively.

5. Performing a Polynomial Fit

- It is often useful to perform a low order polynomial fit to the signal event for example when there is an overall upward or downward trend. Define the order of the polynomial and also the number of iterations required in the text boxes:

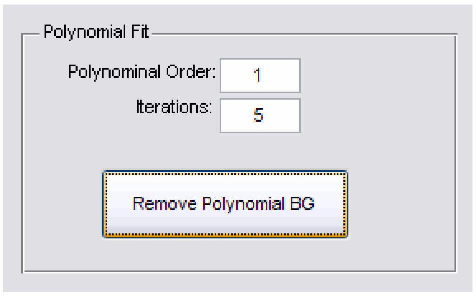


Figure 9. The polynomial fit frame

- Each iteration reduces the effect of the signal peaks on the fitting in the same way the wavelet transform iteration does. See the paper for more information on this technique.
- If a background was imported and subtracted from the signal, the polynomial fit will be performed on the filtered signal that is shown on the axes on the right.
- Once the fitting has finished, the fit is shifted in the y scale so that all data points are larger than zero.
- The background fit on the left hand axes will consist of the polynomial fit plus the bare background whilst the filtered signal will consist of the original signal minus the background fit:

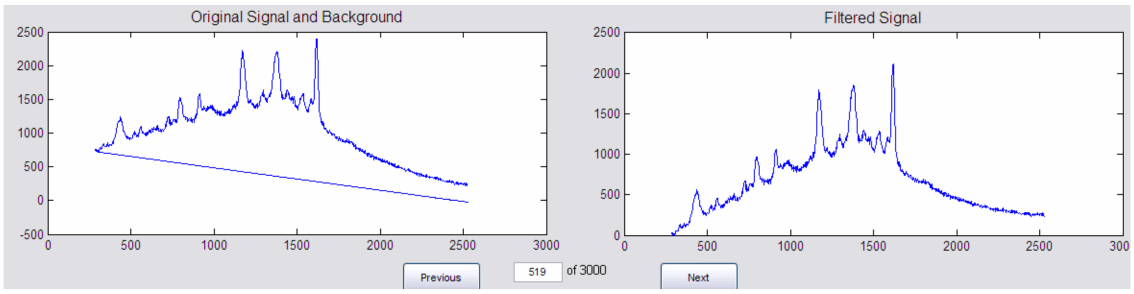
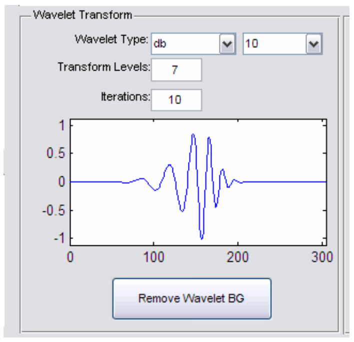


Figure 10. The background fit and filtered signal after the polynomial fit has been performed

- If the polynomial fit is unsatisfactory, click “Reset Single” which will then completely remove the background fit (including the bare background). The bare background does not need to be imported again but can be removed just by clicking “Remove” in the “Bare Background” frame. This will then take you back to the point before the polynomial fit had been made. It is also possible to skip this stage and move straight onto the wavelet transform.

6. Performing a Wavelet Transform

- The wavelet transform is performed on the filtered signal using the wavelet toolbox in MATLAB.
- There are several parameters that must be defined before the wavelet transform can be performed. The first is the wavelet type. Each time a new wavelet type is chosen it will be plotted on the axes in the Wavelet Transform frame:



Wavelet Transform

Wavelet Type: db 10

Transform Levels: 7

Iterations: 10

Figure 11. The wavelet transform frame with the level 10 Daubechies wavelet shown

- We have found that the best wavelet type to use is the level 10 Daubechies wavelet which is shown in figure 11. However, other wavelet types may be more suitable depending on the shape of the signal.
- The second parameter that needs to be defined is the transform level. The higher the transform level, the lower the frequency of the background fit. The best method for setting this value is to start with a low level (leaving the number of iterations at 10), say 3, and click "Remove Wavelet BG". If the background seems to be curving into the signal peaks, click "Reset Single", remove the bare background and polynomial fit and perform the wavelet transform again after increasing the level by 1. Repeat this process until satisfied with the result. The transform level will typically depend on the signal peak widths, the density of the peaks and the number of signal points.
- The final parameter is the number of iterations which is explained in detail in the paper. 10 iterations are typically enough for the fit to converge but this may change depending on the signal to background ratio.
- After this stage the background fit should be complete and the filtered signal should be completely background removed:

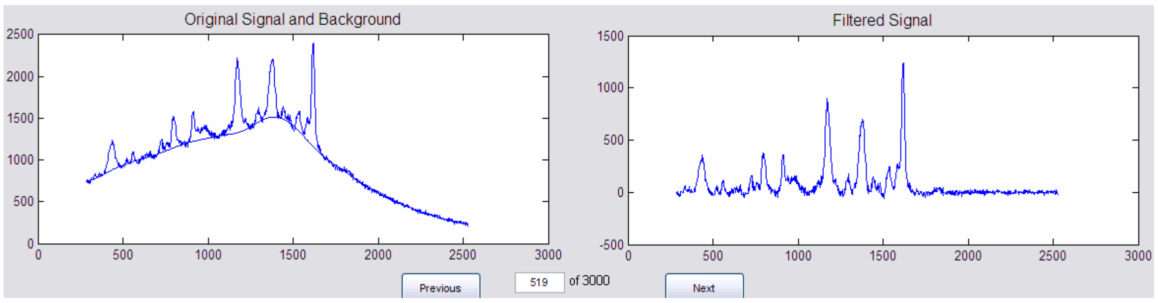


Figure 12. The signal event after a 6 level wavelet transform.

7. **Performing a Fourier Transform**

- Now that the background has been removed, it is often useful to perform a de-noising of the filtered signal. This could be done using wavelet transforms and thresholding, but for speed and flexibility we have chosen to use a low pass Fourier transform. This is done with the MATLAB fast Fourier transform (fft) function.
- The parameters to be defined are the cut-off which is related to the frequency at which all frequencies below are cut out, and the width which corresponds to how quickly the frequency components around the cut-off value are cut out.
- After the "Transform" button is pressed, the Fourier transform and the low pass filter will be shown on the axes:

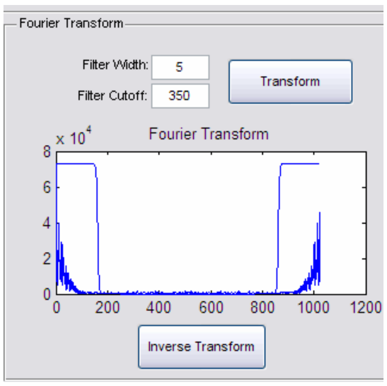


Figure 13. The Fourier transform and low pass filter function for the filtered signal

- The actual signal will not be modified until the "Inverse Transform" button has been pressed and hence the width and cut-off can be changed and the signal re-transform by clicking "Transform" again.
- Once the filter is satisfactory, click "Inverse Transform" and the filtered signal will be modified. If the filtering is not satisfactory then you will need to press "Reset Single" and redo the background removal which should only require pressing a couple of buttons as the parameters will already be set.
- After this step the signal will be completely background removed and de-noised:

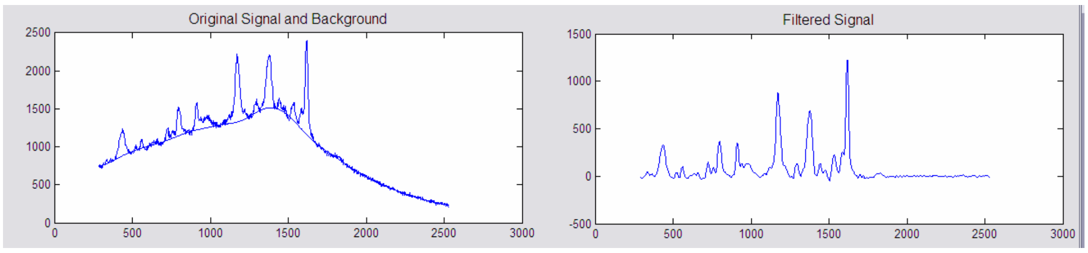


Figure 14. The filtered signal after it has been de-noised and had its background removed

8. Filter all Signal Events

- Now that all of the parameters have been set, we can now perform a background removal and de-noising to all of the signal events. Click "Filter All Events" after selecting which processes are to be applied to every signal event:

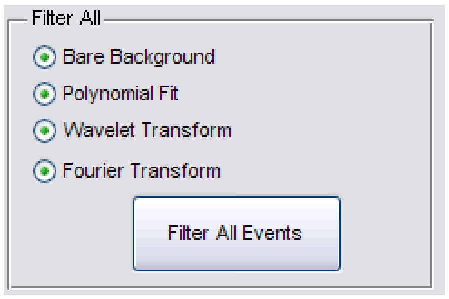


Figure 15. The filter all frame

- This process may take a few hours especially if all of the options are selected and there are many events. The current event number being filtered will be printed in the MATLAB command window to help estimated how long the process will take.
- The signal events and their filtering can be scanned through using the previous and next buttons or by changing the current event number once the filter all process has finished.
- If the results are unsatisfactory, click "Reset All" and make appropriate changes to the initial parameters and re-perform the filter all process.

9. Saving the Data

- Once all of the signal events have been filtered and you are satisfied with the results, you can save the data in a MATLAB file by clicking “Save All”. The MATLAB file can be imported into MATLAB using the “import data” option. The workspace will contain 4 variables, the wavenumber which contains the x-scale values, spectra which contains the original signal data, background which contains the background fits, and filtered spectra which contains the de-noised and background removed signals. These names are artifacts from our research where we look at SERS spectra.
- It is also possible to save just the currently observed event by clicking “Save Single”.