Spectral Analysis User Manual

1. Main Interface

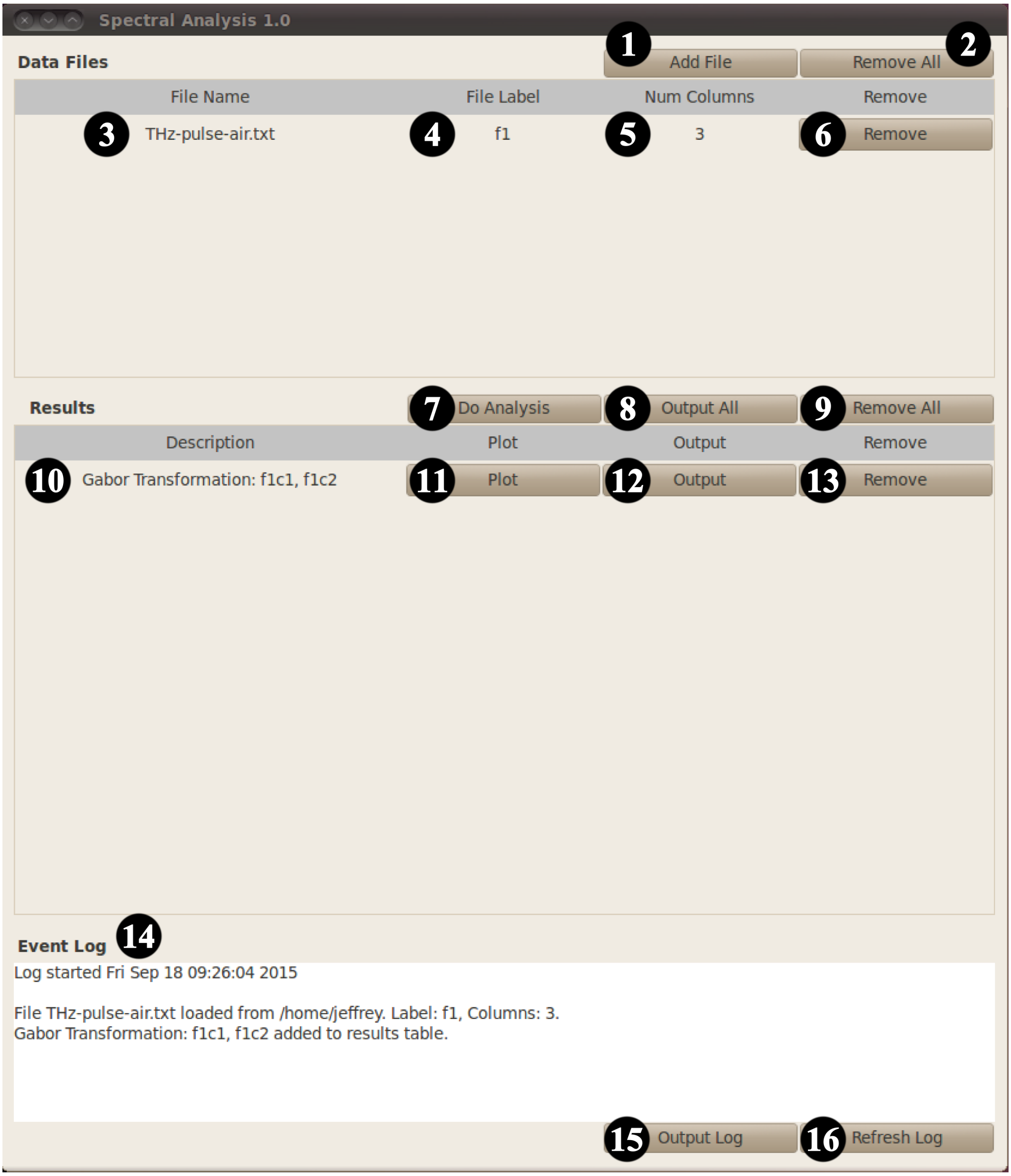

Figure 1. Main window for the Spectral Analysis user interface. Data (file) input and output is managed here, and it is from here that the various analysis features are launched.

If the Data Files dialogue, Results dialogue or Event Log become too full, scrollbars appear. Don't worry - you won't run out of space!

1) Search for a data file to be loaded by the program. Lines started by a % sign are taken to be comment lines, so these lines will not be loaded from the data file. Data is expected to be found in columns delimited by either tabs, spaces or a combination of both.

2) Removes all data files previously loaded from the Data Files dialogue, resetting it.

3) Name of the data file.

4) Label for file. These are used by analysis functions to indicate which data you want to use. Files are labelled f1, f2, f3, ... as each file is added. Even if a file, say f1, is removed, files will continue to be labelled in this order. In other words: the next file added would not receive the label f1; it would receive the next label to be assigned from the sequence, as per normal.

5) Number of columns in the data file. Columns are labelled from left to right as c1, c2, c3, ...

6) Removes the data file in that row, and removes all data loaded from it from the program.

7) Opens up the 'Do Analysis' window (see fig. 2) that allows the user to start carrying out analysis.

8) Allows the user to navigate to somewhere they want all the results created so far to be saved to. Here the program will create a folder using the name the user types into the Output Files window. In this folder, each set of results will be saved to a .txt file. Each .txt file will have the same name as the result's description (see **10)**), with the spaces replaced by underscores.

9) Removes all results from the Results dialogue, resetting it.

10) Description of result. This will often indicate which data (from the Data Files dialogue) was used to create it. For instance, 'f1c2' indicates that the second column of file 1 was used to calculate the result featured. These labels appear in the order you encounter their entry boxes in the data entry window used to generate the result when reading across it from left to right, top to bottom.

11) Creates an interactive plot of the result using the relevant plotting routine. This allows further analysis of the result to be performed via the functions available through the plot window.

12) Choose a location to save a file containing the data associated with that result row. A comment line is added automatically to indicate what the data in each column means. A second comment line will be added if any single-value results were calculated; e.g. the thickness of a sample.

13) Removes the result in that row, and removes all data associated with the result from the program.

14) The Event Log keeps a running record of import actions made by the user that they may find useful to refer to at a later time.

15) Choose a location to save a file containing the event log.

16) Clears all text from the Event Log. The Event Log is then restarted, with the time of the refresh being recorded at the start of the Event Log.

2. Do Analysis Window

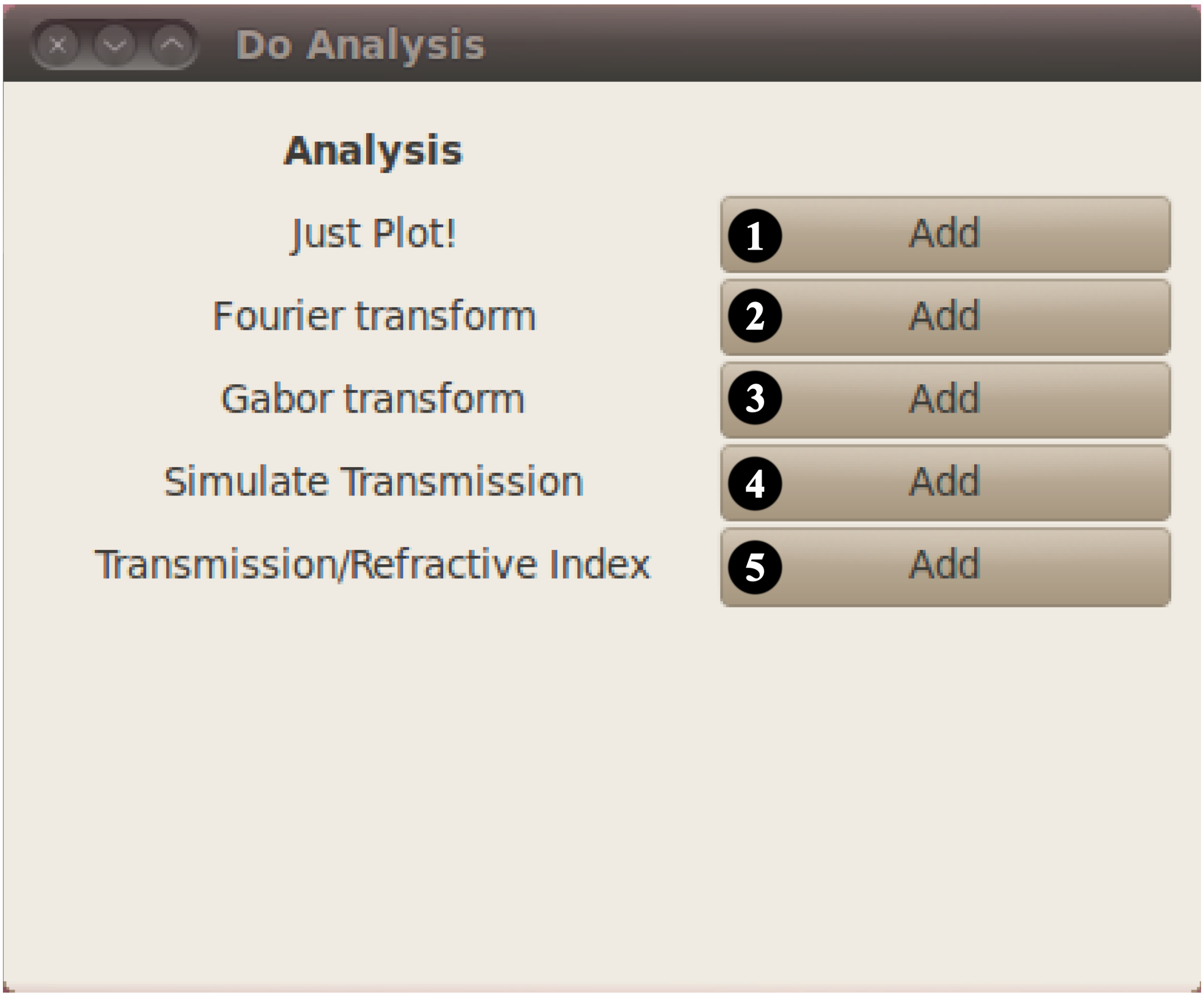

Figure 2. The 'Add' buttons to the left of this window launch the different kinds of analysis that can be carried out.

1) Currently incomplete. Used to create plots with multiple result data sets, etc. on them.

2) Allows Fourier and inverse Fourier transformations. In accordance with optics definitions, the transform and inverse transform have the reverse meanings of those used in the numpy fft library.

3) Gabor transforms a data set. This means that a data set is multiplied by a Gaussian selector window and then Fourier transformed from the time to the frequency domain.

4) Currently incomplete. Allows the user to build up a material stack layer by layer. The transmission of light through the material is then simulated.

5) Takes a time domain pulse of light after it has been passed through a sample and a reference pulse. The program uses these to calculate the transmission spectrum of the sample as well as the frequency dependence of its refractive index. A good model-independent estimate of the sample's thickness can also be made.

3. Fourier Transform Window

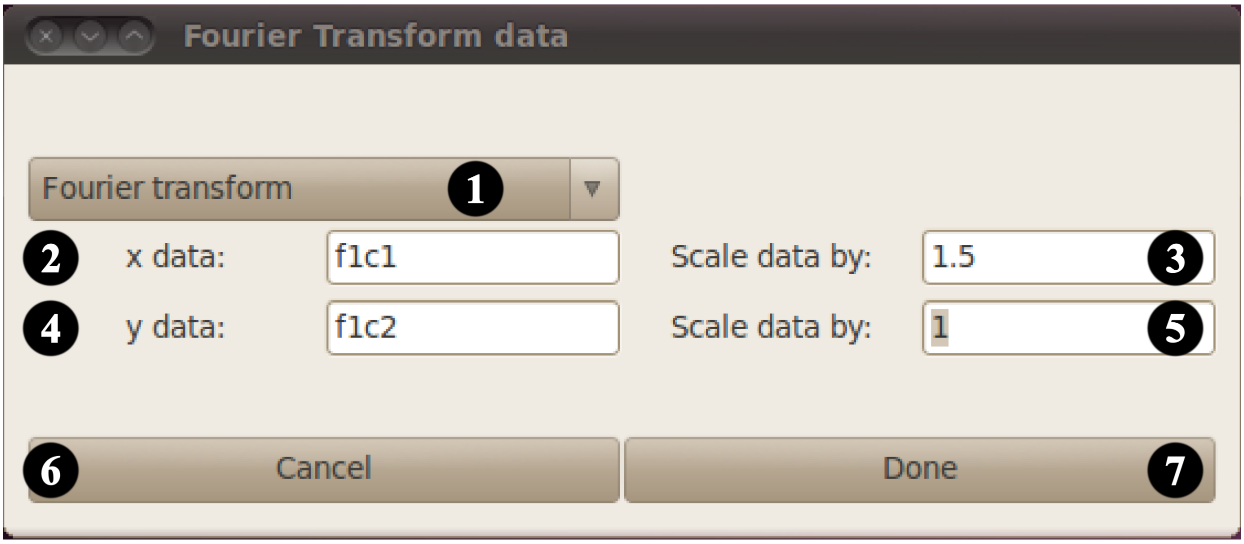

Figure 3. Fourier analysis window.

1) Choose between Fourier transform and inverse Fourier transform. In accordance with optics definitions, these have the opposite meanings to those given in the numpy fft library.

2) Data entry field for the independent variable. For example, time.

3) Allows the data of **2)** to be scaled. By typing a command such as FLIP before the f*c* statement, it is also possible to do things like reverse the order of the data set.

4) Dependent data set; the signal data.

5) Allows the signal data to be scaled.

6) Cancel this analysis. Closes the analysis window.

7) Starts the program performing the analysis.

4. Transmission/Refractive Index Spectra Window

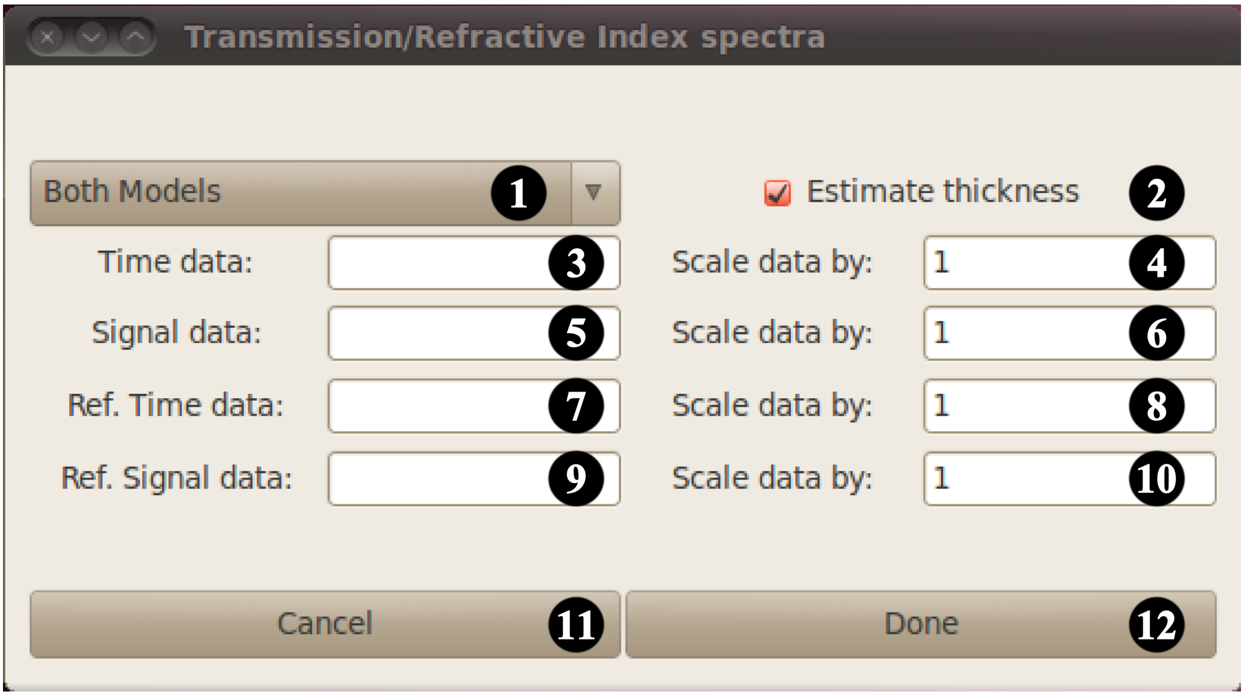

Figure 4. Compares a light pulse that has been passed through a material with a reference pulse to infer the transmission and refractive index spectra of the material.

1) Select how to model the material. The choices are as a thick material, a thin material or to use both models. If the default: 'Both Models' is selected, the program will output the results for both models.

2) If this box is checked, a model-independent estimate of the material's thickness will be found on the second line of outputted results files.

3) Input box for time data for pulse passed through sample.

4) Scales the time data of **3)**.

5) Input box for signal data for pulse passed through sample.

6) Scales the signal data of **5)**.

7) Input box for time data for reference pulse.

8) Scales the time data of **7)**.

9) Input box for signal data for reference pulse.

10) Scales the signal data of **9)**.

11) Cancel this analysis. Closes the analysis window.

12) Starts the program performing the analysis.

5. Gabor Transform Window

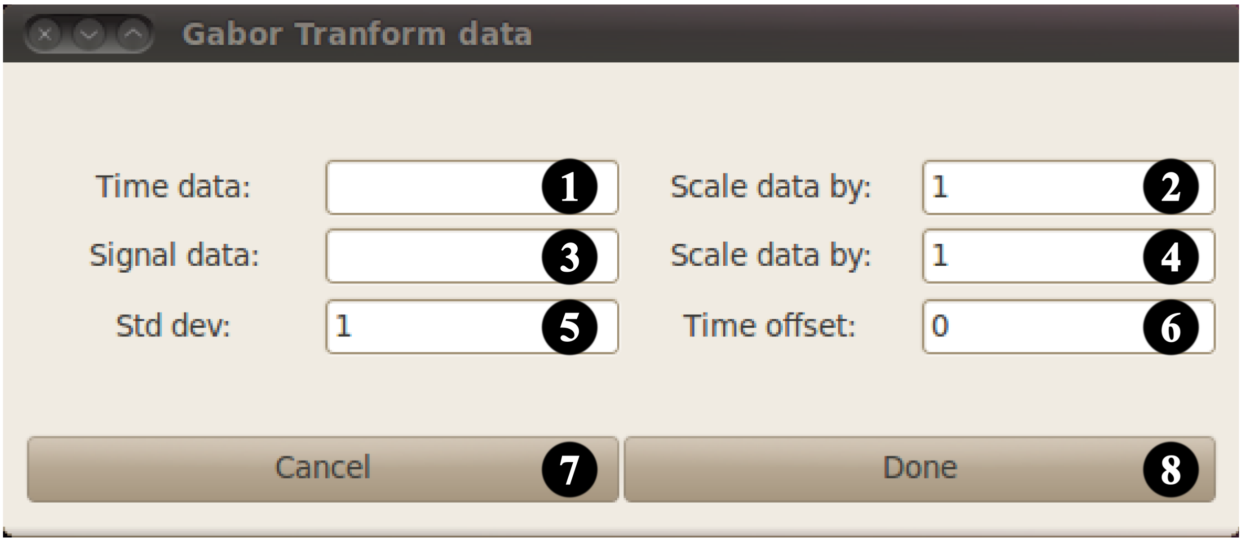

Figure 5. Gabor analysis window.

1) Input box for time data for pulse passed through sample.

2) Scales the time data of **1)**.

3) Input box for signal data for pulse passed through sample.

4) Scales the signal data of **3)**.

5) Standard deviation of the Gaussian selector window in the time domain.

6) Time offset of the Gaussian selector window in the time domain.

7) Cancel this analysis. Closes the analysis window.

8) Starts the program performing the analysis.