# PYTHON SOFTWARE

The software was written using Python 3.10. The Python distribution was an Anaconda version running under Windows 11 using the Spyder IDE. The main program is “LMStemplateCMF.py”. The following Python modules are used: time, sys, os, shutil, ctypes, numpy, matplotlib, openpyxl and PyQt5; as well as the following local modules or classes: CMFcalc.py, CMFplot.py and CMFtemplates.py, which are part of this package.

LMStemplateCMF.py is the main program. The ui files are “user interface” files used by PyQt5 to generate the graphical user interfaces. The directory “CMFs\_in” holds the csv files that contain the discrete data that make up the Stockman & Sharpe (2000) standard. These, which will be described later, are not used in the main program, but may be of use for future developments.

If running from Spyder, set Tools -> Preferences -> IPython console -> Graphics -> Graphics backend to “Automatic”. I run the program in a dedicated console.

To start the program run “LMStemplateCMF.py” from within the running directory using the command line or a development environment such as Spyder. Or add the running directory to the Python path.

This program comes with no guarantees. Use at your own risk. Corrections and improvements welcome.

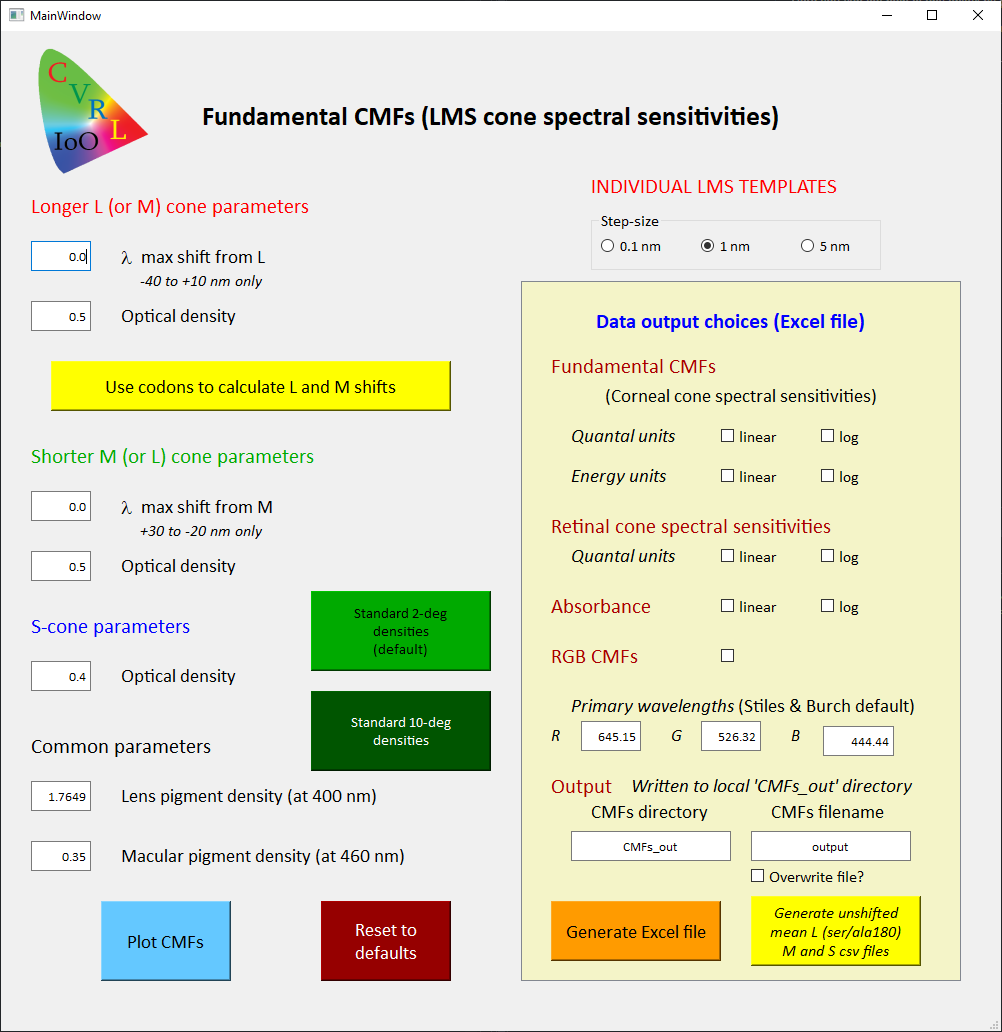


Figure 1: Main window

Initially, the text boxes, radio buttons and tick boxes are set to their default values. The default step-size for the calculations and data output is 1 nm. This should be ideal for most purposes, but step-sizes of 0.1 and 5 nm can also be selected.

The text boxes on the left are editable so that the user can adjust the input parameters. The default parameters are those assumed by Stockman & Sharpe (2000) , so if the user clicks “Plot CMFs” or “Generate Excel file” without making any changes, the output will be the standard, unmodified Stockman & Sharpe 2-deg cone fundamentals.

Under “Longer L (or M) cone parameters” the user can accept the default parameters or shift the max and/or alter the photopigment optical density. Plausible values for those parameters, and the effects they have, were discussed above. The max shifts for LM are limited to ‑40 to 10 nm. Under “Shorter M (or L) cone parameters” the user can also accept the default parameters or shift the max and/or alter they photopigment optical density. The max shifts for ML are limited to 30 to ‑20 nm. If the user exceeds the shift limits for either M or L, a warning is written below the text box. Under “S-cone parameters” the user can accept the default photopigment optical density or alter it. The “Common parameters”, which affect the spectral sensitivities of all three cone types, are the lens pigment density (defined as the density at 400 nm) and the macular pigment density (defined as the density at peak, which is at 460 nm). The default values are again appropriate for the Stockman & Sharpe 2-deg cone fundamentals. The range of plausible values for the macular and lens pigment densities were discussed above. The details within yellow frame labelled “Data output choices (Excel file)” will be described below.

An important feature of the program can be accessed by clicking the button “Use codons to calculate L and M shifts”. This opens the menu shown in Figure 1. In this menu, the user can select which of the seven codons within the “Shorter ML-cone” opsin on the left or within the “Longer LM-cone” opsin on the right are M-cone codons or L-cone codons. Each time a radio button is pressed, the cumulative photopigment shift is recalculated and shown beneath the codon selections. The shifts associated with each change from M to L or from L to M are tabulated in Tables 1 and A1. Note that based on a review of the previous data on the effects of changing codons on max, changes at position 233 or 309 are assumed to have no effect on max. Therefore, clicking the radio buttons for those codons does not shift the photopigments (these codons have been retained for completeness, since some evidence, see Table A1, suggests that there may be small < 1 nm effects).

The underlying M-cone and L-cone absorbance templates differ slightly in shape (see Figure 4). Which one is used to generate a new LM or ML cone fundamental depends on the final max of the shifted pigment. and closes the codon screen Normally, the M-cone template is used to generate the new ML-cone fundamental, and the L-cone template is used to generate the new LM-cone fundamental. However, if the shorter LM-cone pigment is shifted by 14 nm or more to longer wavelengths (so that it is closer to L), the L-cone rather than the M-cone template is used to generate the new fundamental and the message “Using the L-cone template for shorter pigment” appears on the screen. Similarly, if the longer LM-cone pigment is shifted by 14 nm or more to shorter wavelengths, the M-cone rather than the L-cone template is used to generate the new fundamental and the message “Using the M-cone template for longer pigment” appears on the screen.

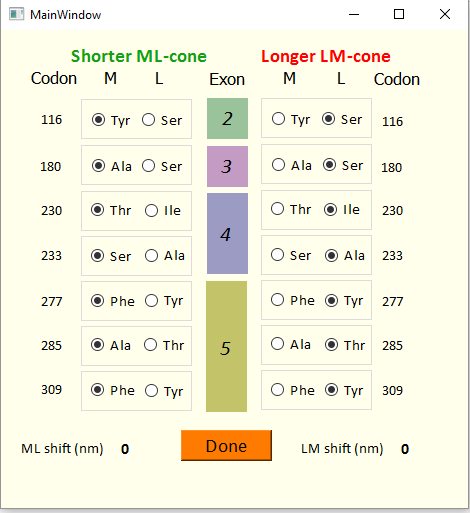


Figure 2. Select M- and L-cone codons

When the codon screen is present, users can press the button “Direct input of L and M shifts” on the main screen to close the codon screen and return to directly editing the spectral shifts in the text boxes. A button “Reset to defaults” on the main screen resets all the text boxes to their default values.

Once the desired parameters are in the text boxes within the main menu, the button “Plot CMFs” can be pressed to generate the plots shown in Figure 3 in a new window. (The types of plots that appear can be modified by directly editing the program.) The four plots in the top left of the window show the newly generated L-, M- and S-cone fundamentals as red, green, and blue solid lines, respectively, all normalised to unity peak. The first plot on the left shows the linear LMS cone photopigment absorbances, the second plot shows the linear LMS corneal cone spectral sensitivities (or cone fundamentals) in energy units, the third plot shows the log10 LMS cone photopigment absorbances; and the fourth plot shows the logarithmic LMS corneal cone spectral sensitivities also in energy units. Energy units are shown because color matching functions, in general, are given in energy units. The differences between the solid and dashed lines give a visual indication of the changes in the cone fundamentals caused by the change in parameters.

The top right panel shows a plot of the *l*,*m* cone chromaticity coordinates for the new fundamentals plotted both as a continuous line and at 20-nm steps (red circles). The *l()* and *m()* chromaticity coordinates are:

 and  [1]

The second panel on the righthand column compares plots of L-M for the newly calculated fundamentals (solid line) and for the unmodified Stockman & Sharpe 2-deg fundamentals (dashed line). This plot gives an indication of the size of the red-green chromatic signal and how it depends on the shapes of L- and M-cone fundamentals. The panel on the bottom row shows the RGB colour matching functions for the wavelengths shown on the main screen (defaults are the standard Stiles and Burch wavelengths)

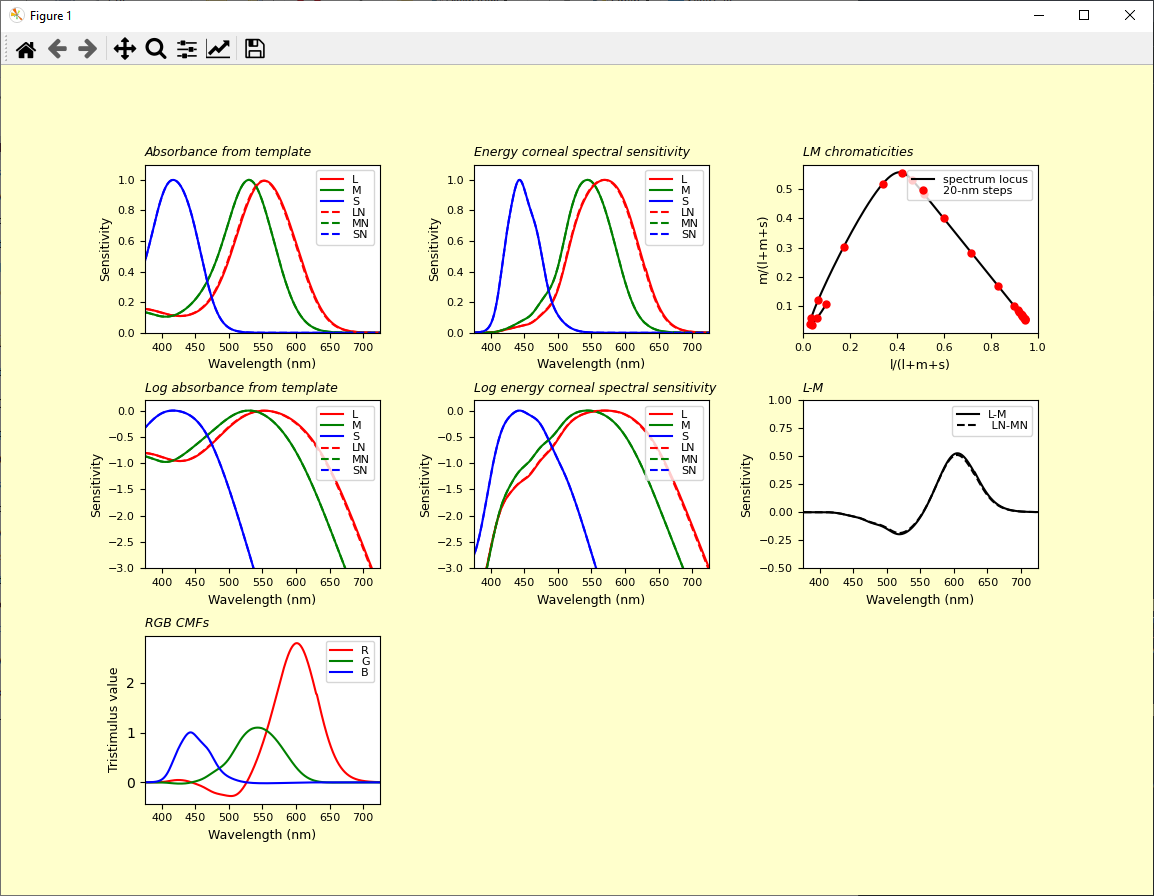


Figure 3. Graphical output screen

The user can repeatedly go back and change the parameters or codons to produce more plots in new windows.

If users want to store the newly generated cone fundamentals as an Excel file, they should use the the yellow frame on the right hand side of the main screen, which is highlighted in Figure 4. There are nine data output options (1-9), each of which can be selected using radio buttons. The new LMS cone fundamentals can be saved in (1) linear quantal units, (2) logarithmic quantal units, (3) linear energy units, or (4) logarithmic energy units. Or the retinal cone spectral sensitivities can be saved as (5) linear quantal units, and (6) logarithmic quantal units. The photopigment absorbance functions can be saved in linear (7) and logarithmic (8) units. Finally, colour-matching functions for a triplet of wavelengths (9) (default is Stiles and Burch’s RGB) can be saved.

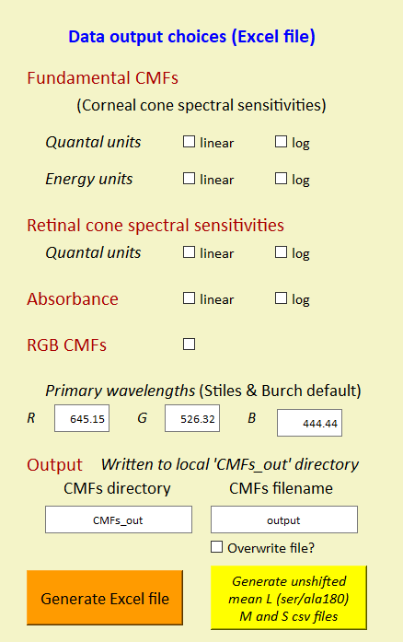


Figure 4. Data output selections (in Main window)

The “*CMFs directory* entered into the editable text box at the bottom is used to write a subdirectory of the current directory. The data is saved within that directory with the name in the “*CMFs filename*” editable text box. A warning will be given if a file with that name already exists and will not be overwritten unless the *Overwrite file* checkbox is checked.

The contents of the “data\_in” directory are shown in Figure 5. They are comma-delimited value ascii files, copies of which can also be found on the CVRL website. The naming convention is:

“\_fine”, “\_1”, “\_5” signify data at 0.1, 1 and 5-nm steps, respectively.

“ss” signifies Stockman & Sharpe (2000).

“2\_10” signifies 2-deg fundamentals calculated from the 10-deg fundamentals (the standard 2-deg functions are so calculated—see above)

10 signifies 10-deg cone fundamentals calculated as a linear transformation of the Stiles & Burch (1959) 10-deg CMFs.

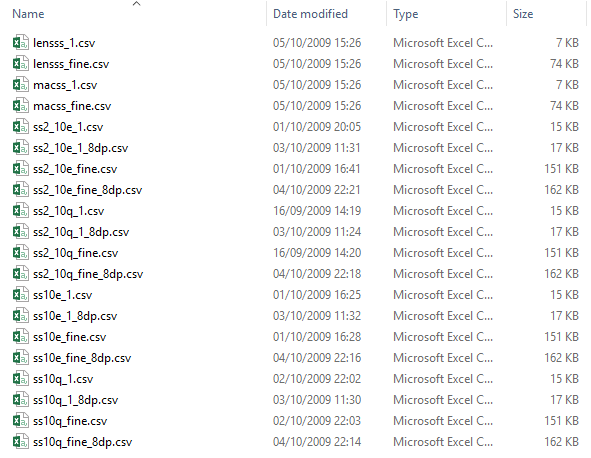


Figure 5. Data input csv files

“q” and “e” signifies data in either quantal or energy units, respectively.

# SOFTWARE COMPONENTS

For more details, see the comments in each file. The \*py and \*.ui files are ascii.

LMStemplateCMF.py:

Main program file.

CMFcalc.py:

Module that contains the functions for cone fundamental calculations, including changes from corneal to retinal sensitivities and back, changes to retinal absorbance and back, and energy to quantal conversions and back. Called by LMStemplateCMF.py.

CMFplot.py:

Plotting class that uses pyplot from matplotlib to generate various plots. Called by LMStemplateCMF.py.

CMFtemplates.py:

Module that contains the functions for calculating the template shapes for cone fundamentals, and macular and lens density spectra. Called by newCMF.py.

inputCMFgenerate.ui:

User interface file used by PyQt5 to generate the main screen graphical user interface. Can be edited with Qt designer or equivalent.

LM\_codon.ui:

User interface file used by PyQt5 to generate the codon screen graphical user interface. Can be edited with Qt designer or equivalent.