**Instructions for Running the Code:**

1. Place the raw (should be CSV files) generated by the Lumicycle software into a folder.

2. Run the code.

3. Click the 'Open file' button and select the CSV file you want to analyse.

4. The code will automatically detect peaks and troughs in the data and attempt to save this information in a new Excel file. A dialog box will prompt you to save the Excel file, and a separate window will display a plot of the original data, with all peaks and troughs labelled.

5. Once the Excel file is saved, the code will open it automatically. The data contained in the Excel file includes: Peak/Trough, Number of Peak/Trough, counts/sec, Time (Day), and Phase. Counts/sec and Time (Day) correspond to the original CSV data. Phase is calculated by: [Time (Day) – the integer part of Time (Day)] \* 24.

6. Remove the data not wanted by referring to the plot.

7. Copy and paste the processed data into the template.

8. The plot will be automatically saved in the same folder as the CSV files, with the filename 'plot\_csv file’s name'.

9. It will be better to close the plot window before continuing. To analyse the next file, repeat the process by clicking 'Open file'. To exit the program, click the 'Exit' button.

**Tips:**

1. Ensure that all required Python packages are installed before running the code (pandas, tkinter, scipy, matplotlib, os, platform, subprocess).

2. Avoid resizing the plot window, as this may cause the plot to disappear from view (this won’t affect the saved plot).

3. Throughout the execution, the Python console will give several notifications in console for the executed actions, including:

- File Pathway: Indicating the opened file.

- Data for peaks and troughs being found.

- Data saved: Notification that the data has been saved into the Excel file.

- Plot saved: Notification that the plot has been saved.

- Exit: Confirmation of the end of the program.

4. The crucial function of this program is `find\_peaks()`, which identifies points where the monotonicity of data of count/sec’s changes, thus, sometime it may be over sensitive. To address this problem, consider excluding such 'errors' manually by referring the plot, which is a safer way. For code-based solutions, refer to online resources about `find\_peaks()` to see how to set a threshold for detection.