LICOR Data Processing Tool Manual

Table of Contents

[Introduction 2](#_Toc94882677)

[Using the tool 2](#_Toc94882678)

[Software requirements 2](#_Toc94882679)

[Required files 2](#_Toc94882680)

[Raw LICOR data 2](#_Toc94882681)

[Field data 2](#_Toc94882682)

[Running the tool 2](#_Toc94882683)

[Cut Methodology 4](#_Toc94882684)

[Viewing your results 4](#_Toc94882685)

[Editing the script 5](#_Toc94882686)

[Software requirements 5](#_Toc94882687)

[Building executables 6](#_Toc94882688)

[Support 6](#_Toc94882689)

# Introduction

The LICOR data processing script is a tool created in Python that enables users to easily and quickly obtain formatted gas flux concentrations from the raw data outputted by the LICOR trace gas analyzer.

# Using the tool

## Software requirements

The script executable and source code can be found in the folder “LICOR script” under the files for the general channel of the Wetland Soils and GHG Exchange Lab Microsoft team.

Within this folder, select the appropriate executable depending on your computer’s operating system.

There are no software packages that need to be installed in order to run the tool.

## Required files

The LICOR tool uses two files, the raw data outputted by the LICOR itself and a user created field data file containing each flux’s name, start and end times and start and end temperatures.

### Raw LICOR data

Obtain your run’s text file from the LICOR itself. Leave the contents of the file unchanged.

### Field data

In the “LICOR script” teams folder, you’ll find a template for the field data .csv:

Graphical user interface, application, table, Excel

Description automatically generated

Save a copy of this file as a .csv and add a new row with relevant information for each flux measured by the LICOR. You can add other columns to this data, as long as the first six columns stay unchanged.

## Running the tool

Once you have the LICOR raw data and the field data .csv, double click the executable to run the tool. You should see the following window appear, as well as a file explorer asking you to select your field data file:

Graphical user interface

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

Once you’ve selected the field data file in the file explorer, another file explorer should open up asking you to select the raw LICOR data:

Graphical user interface

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

Once you’ve selected the raw LICOR data, the tool window will prompt you for information used to generate the final report:

Graphical user interface, text

Description automatically generated

After entering this information, a graph window will appear displaying the first flux from start to finish. Within this window you can ‘prune’ bad data, whether that’s an ebullition event or sensor inconsistencies. Use the mouse to drag the two red lines to surround the region you want to cut, then press enter. If you want to undo any cuts, press ‘r’ to reset the flux. Once you’re satisfied with the pruned flux, use the arrow keys to move onto the next flux. Shown below is a flux before and after a cut.

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

## Cut Methodology

Since the desired information is the linear rate of change of the gas concentration, cuts made in the middle of a flux use constant offsets to best identify this linear relationship. If a cut is made in the middle of a flux, it shifts all of the concentrations after the cut by a fixed offset calculated as the difference between the last concentration before the cut and the first concentration after. In addition, all data points after the cut are shifted ‘back in time’ by the width of the cut. The result of this is connecting the two points immediately before and after the cut, so as to pretend the region in the cut ‘never happened’, therefore attempting to preserve a straight line in the data.

If this is more data manipulation than you’re comfortable with, limit your cuts to only those encapsulating the edge of the flux, instead of in the middle of the data.

Diagram

Description automatically generated

## Viewing your results

Once you’ve finished with each flux, the tool will prompt you to save your results as an excel spreadsheet. On the first sheet you’ll see a summary of all fluxes, showing the calculated rate of change and overall flux for each. As well, the total data lost to cuts will be displayed as a percentage of the original data that has been cut.

Graphical user interface, application, table, Excel

Description automatically generated

In addition to the summary sheet, there will be one generated sheet for each flux displaying a more detailed breakdown of the results for that flux. These sheets will display their flux’s original data sets and the final data sets after the cuts. If there’s been an offset applied to any entry in the original data sets, the value of the offset will be displayed in that row.

Finally, each individual sheet in the results will display a chart indicating which portions of the data has been cut/shifted. The data displayed in red has been cut, while the data in green is what’s been used to calculate rate of change and flux.

Graphical user interface, application, table, Excel

Description automatically generated

# Editing the script

## Software requirements

The script was written using Python 3.10.1, with matplotlib and XlsxWriter as the only non-native library. Python 3.10.1+ can be installed for any operating system at the following link:

* <https://www.python.org/downloads/>

Once Python is installed, matplotlib and XlsxWriter can be installed with the following command in a terminal session:

* Pip install matplotlib
* Pip install xlsxwriter

Once this environment is set up, the source code file LICOR.py’ can be edited in your preferred text editor and ran using the following terminal command in the same directory as ‘LICOR.py’:

* Python LICOR.py

## Building executables

Once you’re finished editing the script, a new executable can be built using the python library pyinstaller. The executable will enable anybody to use the new script, even if they don’t have Python installed. Pyinstaller can be installed with the following terminal command:

* Pip install pyinstaller

With pyinstaller installed, a new executable can be built with the following terminal command in the same directory as ‘LICOR.py’:

* Pyinstaller –-onefile LICOR.py

Once this command has finished running, your executable can be found in the folder called ‘Dist’ in the same directory where ‘LICOR.py’ was located.

**Note: Executables built on MAC will only run on a MAC, executables built on Windows will only run on Windows**

# Support

If you have any questions about how the script works, any changes you’d like to see, or any difficulties running the script, please contact me at the following email address:

* btnewton@uwaterloo.ca