**One Click to Uniform Tabular Format**

This ReadMe is written for **version 0.9.2**.

Last updated: 12. July 2019

“One Click to Uniform Tabular Format” (OCUTaF) is a data analysis software designed for promoter activity assays as measured with our robotic systems. It assumes the measurement of an optical density representing cell count (e.g. OD600) and one reporter, either fluorescence or luminescence. OCUTaF reorganizes data in a single file (or rather single files) and performs basic mathematical operations as background subtraction of blank values or calculation of relative reporter units.

## Set Up

There are two ways to start the software:

1. **From EXE file:** Simply double-click the .exe file. A black console window will open, indicating the loading of different modules. Shortly after, the main window will appear.
2. **From Python source files**: Start the software by navigating the console to the directory of the python files. Start with the command “python start\_gui.py”. This requires Python version 3.6 or higher as well as the packages pandas, openpyxl, and xlrd.

For the **help menu**, make sure to have the help folder placed in the same directory as the software.

## Data Types and Usage

OCUTaF software accepts three types of data: Tecan Robot output files, Hamilton Robot output files, and Tecan reader Excel files. They required layout of the files is **highly specific**. Functioning example files can be found in the “example\_data” folders. A brief document with important additional information can be found in the respective folders.

OCUTaF automatically determines which data type it is encountering. It uses **file extensions** to determine which data is input: .asc is interpreted as Tecan robot, .xlsx as Tecan Excel, and .xls as Hamilton robot.

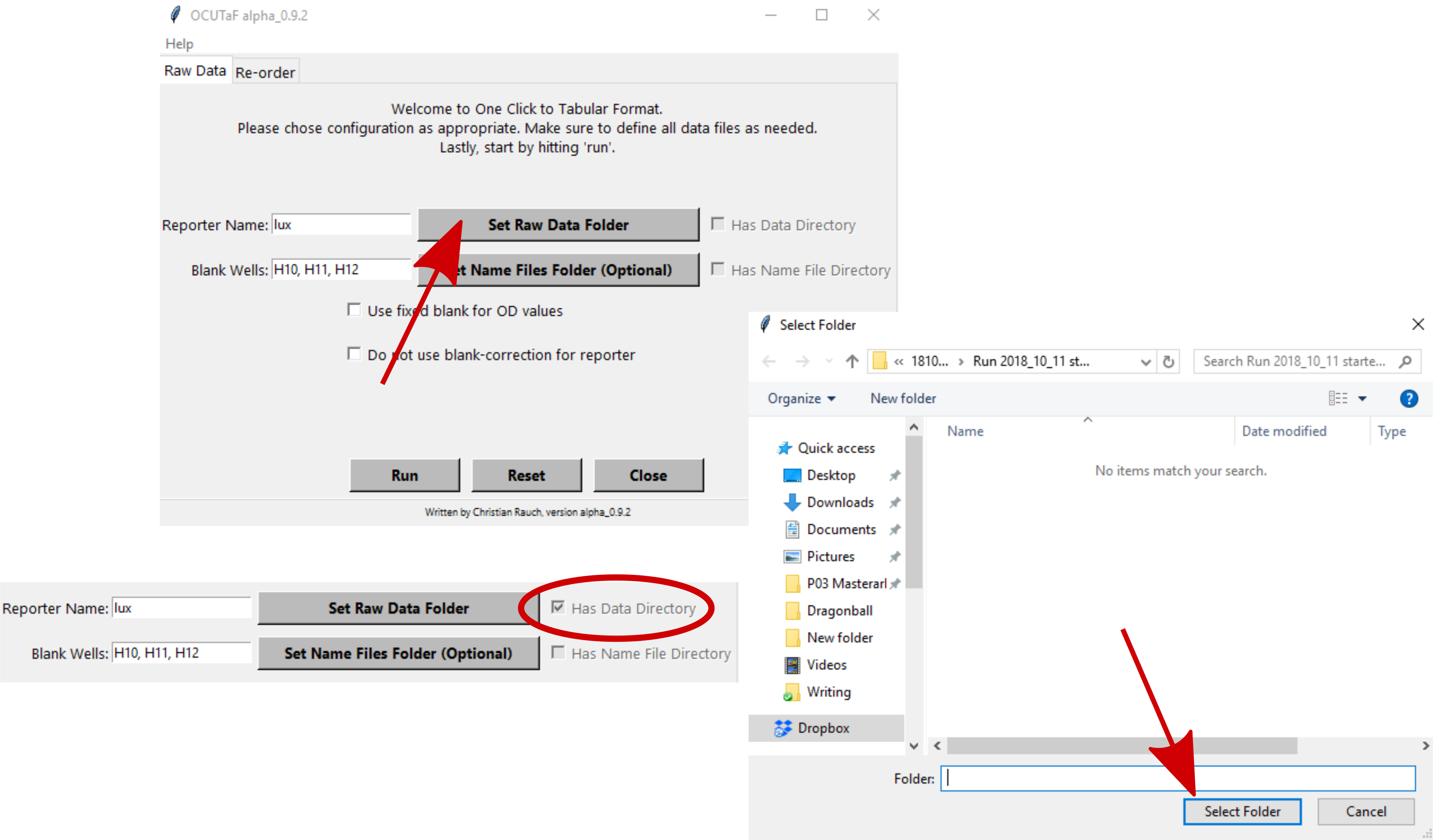
In the names of the input files, a **barcode** for each plate **is expected**. See the specifications in each “example\_data” folder for details.

To **read data**, click on “Set Raw Data Folder”. A navigation window will open. Select the directory in which you raw data is located. OCUTaF will indicate with a checkmark that a folder has been selected (this will also be the case for an invalid selection, for example a directory without any data! This only indicates that ANY directory has been chosen). The folder can appear empty; this still works fine.

Additionally, a **naming file folder** can be specified. For this, see below.

The **reporter name** is purely informative and will be used for naming output files.

**Blank wells** are used for background subtractions. They can be specified in three ways: As single wells, seperated by a comma (H10, H11, H12). As rows of wells (H1:H3 is interpreted as H1, H2, H3). as a full column; for this, only enter a number (for example, 12 will be interpreted as A12, B12, ..., H12).



## Output Files

Several output files are created through OCUTaF. They are all stored in directory containing the raw data. The analysis of the data occurs in three steps:

1. **Simple copying** of input data into a uniform format. This step resembles a Perl script written by Stephan and produces a tab-stopped “.txt” file with raw data of OD and reporter. This generates one file per recognized barcode.
2. **Background-correctio**n of OD and reporter raw values (if not excluded) using the input blank wells and **calculation of relative reporter units**. This creates two files per recognized barcode: One for blank corrected OD and one for relative reporter units.
3. Optional **naming** step. If name files are provided, two files per recognized barcode are created. These files are the same as in step 2, except the well names are replaced by meaningful names. If more than one barcode was encountered, two additional files are created that contain **all OD and all relative reporter data** from the run with their attached names.

## Optional Naming

Naming is an optional step that can speed up further data analysis. The files required for naming are CSV files generated from the German Version of Excel; the seperator is the semicolon ‘;’ character (per default). Name files have to be supplied in a separate folder containing ONLY the name files. Recommended naming would be “plate\_1”, “plate\_2”, ..., to ensure the same alphabetical / lexicographical order as the barcodes. This is necessary for the correct matching of barcodes and name files. For each plate (and therefore each barcode), one name file has to be supplied. The layout for the name file has to be strictly as follows:

1. 2 3 4 5 6 7 8 9 10 11 12

A ... ... ... ... ... ... ... ... ... ... ... ...

B ... ... ... ... ... ... ... ... ... ... ... ...

C ... ... ... ... ... ... ... ... ... ... ... ...

D ... ... ... ... ... ... ... ... ... ... ... ...

E ... ... ... ... ... ... ... ... ... ... ... ...

F ... ... ... ... ... ... ... ... ... ... ... ...

G ... ... ... ... ... ... ... ... ... ... ... ...

H ... ... ... ... ... ... ... ... ... ... ... ...

where “...” is the name of for the respective well.

If the Re-Order function is to be used, names must conform to the style “[construct], [condition]” (without “”). For example, if the expression of Gene A is induced with several concentrations of an inducer, its name could be “Gene A, 50 µM”.

## Mathematics and Statistics

Blanks

Blank correction

Relative reporter units

## Reorder Functionality

Simple Modifcations in the Source Code

- make TSV thingy modifiable

- make default values for blank wells and reporter a thing