How-To file

# for program:

Machine\_Screener\_ver7

# Setup with Python

The program runs a Python script which requires that Python3 is installed on your machine. This can be downloaded for free from [https://www.python.org/downloads/].

# Input data

The program needs as an input a .csv file that contains a column of abstract IDs, a column of abstract texts, a column of gold-standard human tags, and a column of gold-standard human assignments (“IN” or “EX”). There can be other columns present in the file, these will be ignored. The top row of the file should contain headings not data, as this row is discarded during processing.

To save an Excel file as csv, navigate to the sheet of interest using the tabs along the bottom. Click File, Save As, then from the drop-down menu select "CSV (Comma delimited)(\*.csv)". Give the file a name (e.g. YourFile.csv) and click Save.

YourFile.csv must be saved or moved into the same folder as the "Machine\_Screener\_ver7" file. Please copy the Machine\_Screener\_ver7 file into your own folder, and delete it after use.

# Running the program

## Input data, testing for duplicate abstracts

Right click on Machine\_Screener\_ver7, select Open With and then Python. The black command line box should appear and prompt you to enter the name of your training csv file, followed by the columns in which the ID numbers, abstract text, gold-standard screening tags, and gold-standard assignments can be found.

The program will then perform a quick check for duplicate abstracts and will display any duplicate pairs that it finds. This will not include abstract entries that are empty. You can then select whether to continue with the program.

The program will then check within the duplicates it has found for any duplicates that have conflicting human tags. It will display any pairs that it finds. Again you can then select whether to continue the program.

## Checking tags and assignments

The program will then check whether the tag assignments are consistent, and will flag up if they are not. For example, if the tag “BACKGROUND” has been assigned as “IN” sometimes and “EX” at other times then this will be flagged and will have to be corrected before continuing.

## Calculating recommended tags and IN-EX assignments

The program should then run all the way through, taking between 30 seconds and 5 minutes. It will detect any articles that are currently mark as excludes that it think might be includes.

## Output files and internal testing

As it finishes the program will output two files. The file name of both will start with the same identification number that is unique to each time the program is run. The “suggestionFile” contains the articles that have been recommended for review, while the “keyFile” contains additional information that will be of interest after the screening project has been concluded.

The purpose of outputting two files is that the suggestionFile contains a mixture of true recommendations and random ones. Once the recommendations have been reviewed and any changes to IN-EX assignments made, the keyFile can be opened to examine whether the true recommendations were useful relative to the random ones. If an equal proportion of changes were made for true and random suggestions then the program has no utility above chance.

## Troubleshooting

If the program is given data it cannot handle it should flag this up before it quits.

There is some kind of bug where when opening the program it opens, asks for input data, and then immediately closes without an output. I don’t know why this occurs, but there is a way around it.

* Right click on the program, click ‘open with’, ‘choose another app’, then select ‘python’ and tick ‘always use this app to open .py files’.
* The program will run, and close at the same point as before.
* However, now if you double-click on the program back in the folder it should run properly.