User Guide for Alveograph Exporter Software

Written by: Nicholas Sixbury

USDA-ARS Manhattan, KS

September / 2024

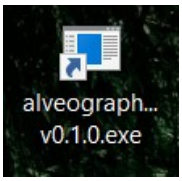
The alveograph is a machine used to measure various properties of dough made from cereal samples. It outputs this information digitally as a txt file with data for each sample. These txt files contain information about various curves, as well as average measurements for one sample, but this information is not necessarily in a format that is easily separated from the rest of the file due to non-standard formatting.

The Alveograph Exporter software was written at USDA-ARS of Manhattan by Nicholas Sixbury in the fall of 2024, developed for use by scientist Rhett Kaufman and his team. The purpose of this software is to take txt files from the alveograph, extract the useful information from each file, and then output that information as a summary in an excel file.

How to use the Alveograph Exporter software:

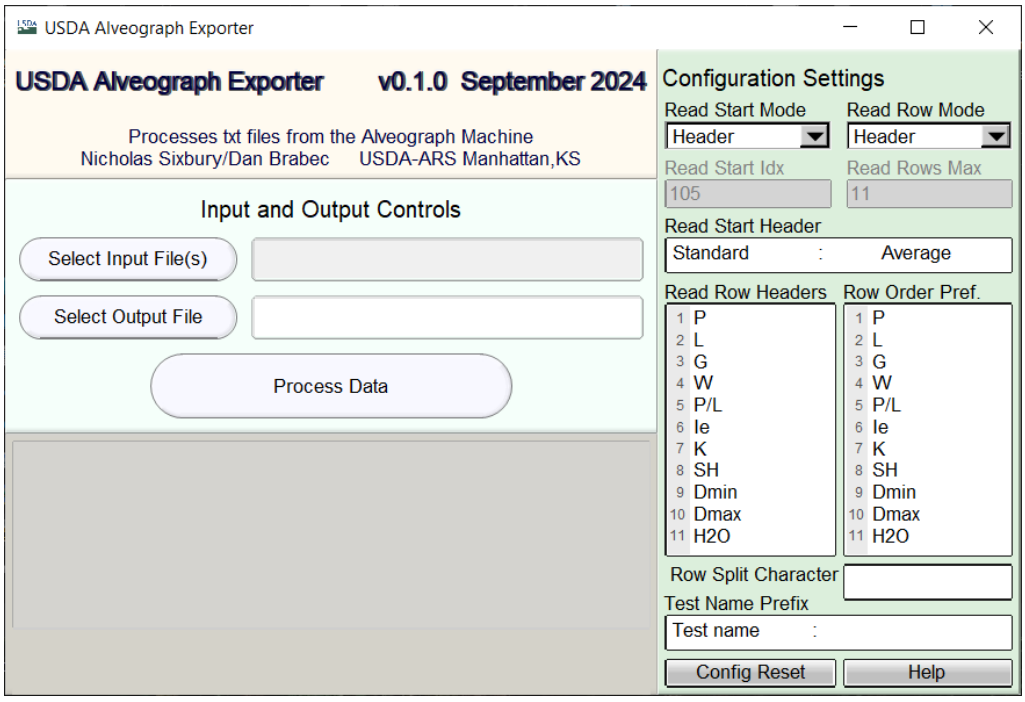
1. **Open the program** from the desktop by using the mouse to click on the icon.

Figure 01: example of shortcut to software



2. Ensure the configuration settings are set as desired. You can click "Config Reset" to reset all settings to the default.
3. Choose input file(s) by clicking "**Select Input File(s)**". You can select as many txt files from the alveograph as you like.

Figure 02: example of program interface



4. Choose an output file name by clicking "**Select Output File**" or just typing a name in the box.
5. Click "**Process Data**" to process the files you selected. After processing has finished, a dialog will appear, asking if you'd like to open the folder where the output file is located. Selecting "yes" in this dialog will show you directly to the output you generated.
6. Once the program has finished processing the file, the section labeled "Input and Output Controls" will be cleared, and you can continue processing as many files as you like.

Advanced Settings Configuration for Alveograph Exporter:

For an in-program reference on all the configuration settings, you can click the "Help" button.
In order to understand the options for configuring the program, it's probably best to start by explaining more of how the program works. In each file, the program generally tries to find two things:

- The test name for the data in that file
- A few rows of data showing the average results for that file

Each configuration setting will be described below. If you would like an example of a working config and file, use the default config (obtainable at any time by clicking "Config Reset") and the sample_input.txt file that should be present wherever your application is installed, in the same directory as the executable file and the help.html file.

Test Name Prefix:

In order to find the test name, the program searches for a heading that goes in front of the name. For example, it might appear as "Test name : 23SC006" or something similar. We want to extract the "23SC006" from that line, so we set the "**Test Name Prefix**" to "Test name : ". The program looks for a line that starts with the prefix and then assumes that anything after the prefix is the test name.
You can easily set the test name prefix by copying the value out of your input file and pasting in the box in the configuration settings labeled "**Test Name Prefix**".

Reading Data Rows: Finding the Start Header:

In order to find the rows of data in the program, we need to find the line above where those rows start. This is referred to as the "**Read Start Header**", and there are two different ways of finding it.

If the "**Read Start Mode**" is set to "**Header**", then you can paste the header you're looking for into the box labeled "**Read Start Header**" in the configuration section. The program will then search each file for a line exactly matching the text you provide, and then look for rows of data underneath that header.

Alternatively, if the "**Read Start Mode**" is set to "**Index**", then you can provide the line number of the row start header. Please note that **the index for the start header uses 0-based indexing**, so if the line numbers in your file start with 1, subtract one from the line number displayed from the start header to find out what you should input into the box labeled "**Read Start Idx**".

Reading Data Rows: Selecting and Parsing the Data Rows:

Once the program finds the Read Start Header, it needs to know which and how many rows to read. In addition, it needs a bit of information on how each row is formatted. In general, it is assumed that each row contains:

- A header for the row, such as "P" or "L"
- A decimal value, which is the actual data for the row
- A character or string separating the header from the value, such as a tab character

It is also assumed that the header comes first, followed by the separation character, followed by the actual value.

The separation character might also be repeated after the value, in which case anything after the second separation character will be ignored.
You can specify the separation character by pasting the text from your file into the box labeled "**Row Split Character**".

If the "**Read Row Mode**" is set to "**Header**", then in order the find the data rows, the program will match the rows it finds against the "**Read Row Headers**" setting. In the "**Read Row Headers**" box, several row headers are provided in a particular order. The program then looks for rows with headers exactly matching the ones provided, in the same order as provided. When the program either runs out of specified row headers or encounters a line which doesn't start with the next header in the "**Read Row Headers**" list, it will stop reading rows for that file.
You can configure the "**Read Row Headers**" by simply typing into the box labeled "**Read Row Headers**". Be careful to type each header exactly, in the same order they appear in your file. Do not include separation characters such as tabs or empty lines at the end of the box. The rows are labeled with line numbers so you can see exactly how many lines are listed.

Alternatively, if the "**Read Row Mode**" is set to "**Max**", then you may simply tell the program to read an arbitrary number of lines after the Read Start Header.
For example: if you wanted to read 11 data rows that occur right after the start header, you could simply enter "11" into the "**Read Rows Max**" box. This is useful if the data rows might have different headers in different files, if the headers are not in the same order, or simply if you want a simpler way of configuring this setting.

Row Order Preference:

There might be a case in which you prefer that your output be formatted with each header in a particular order. The "**Read Row Headers**" specifies how the rows are ordered within your file, but that order may not be under your control. This you can use the "**Row Order Pref.**" box to change this. You can add Row Headers to the "**Row Order Pref.**" box using the same formatting as the "Read Row Headers" box, but Row Order Preference is much less picky. Any rows with the headers provided but not found will simply be ignored. Any row headers found by not provided will be placed at the end of the output, in whichever order they appear.
Thus, "**Row Order Pref.**" will not add or remove any data from your output; it will only rearrange things.