**EXAMPLE FOLDER INSTRUCTIONS**

To help users with the basics of running their workflow steps through the command line interface, we recommend running the example scripts in the example folders (at [https://github.com/rcfgroup/gc-automation](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Frcfgroup%2Fgc-automation&data=02%7C01%7Cmjw77%40leicester.ac.uk%7C5755eaebcb7243596dfe08d81cf52b3a%7Caebecd6a31d44b0195ce8274afe853d9%7C0%7C0%7C637291186603018602&sdata=a9cxHXPVlS%2B%2BiNhMZ6NbXA4ENsflxu7rCcd80XN%2B1IU%3D&reserved=0)). These instructions and examples allow the user to become familiar with the format and check for errors before moving onto more complex custom tasks.

**Before you begin**

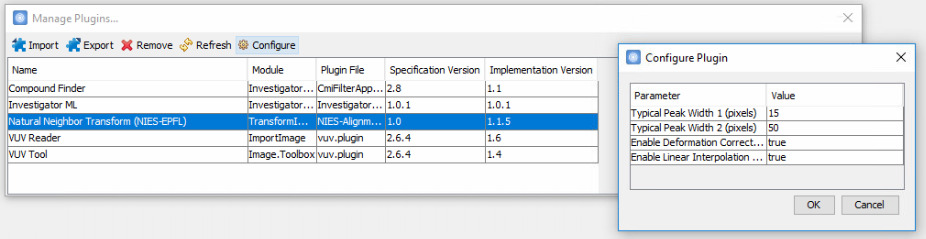
The commercial and free and open-source software (FOSS) for your own custom workflow would already be installed. However, for this specific example, the following steps should be followed to ensure the example folders work as described.

Installing commercial software

Ensure a copy of the commercial software (in this instance GC Image) is installed (the examples were generated in version 2.8r2). A free trial of the software is available on request from <https://gcimage.com/gcxgc/trial.html>.

Installing plug-ins

The examples describe interfacing a published alignment algorithm.1 This algorithm is freely available as a plug-in at <http://gcimage.com/forum/viewtopic.php?f=5&t=104> via the website under Plug-ins and must be installed in order for the second and third example folders to work. To access the plug-in, login to your (free) user account. The original Matlab tool is available at <https://github.com/jsarey/GCxGC-alignment>. Once the plug-in file has been downloaded and extracted to the GC Image program folder, open GC Image, go to ‘Tools’ in the menu bar and from the list select ‘Manage Plugins’. Click the ‘Import’ button and locate and import the plug-in file. The ‘Natural Neighbour (NIES-EPFL)’ plug-in should now appear in the list of imported plug-ins. Click ‘Configure’ and change the parameters to match those below. Click ‘OK’ and close the program.



Installing FOSS

Ensure a compatible version of the free or open-source programming software (in this instance Python) is installed along with the packages used in the script (in this instance the ‘click’ and ‘pandas’ packages). Python is freely available at <https://www.python.org/downloads/>. If using a computer operating Windows 10 or later, you can download the Python app from the Microsoft Store.

To check if Python has been installed, in Windows press  + R and type ‘cmd’ (without ‘’ marks) in the Run window. This opens the command prompt window. Type ‘python’ and press enter. If Python has been installed it will come up with the program details as shown in Figure 1.

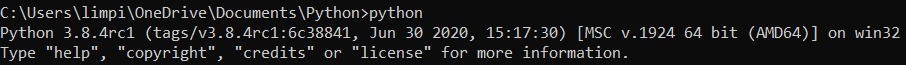


Figure 1: Checking the Python installation in the command prompt window.

To install the necessary packages, check if the package management system ‘PIP’ is installed. To do this in Windows, press  + R and type ‘cmd’ in the Run window. This opens the a command prompt window. Next type ‘pip -V’. If this returns an error, PIP has not been installed. To install PIP, go to <https://bootstrap.pypa.io/> and download the ‘get-pip.py’ file. Save the file to a known location. In the command prompt window, type ‘cd’ followed by the location of the get-pip.py file (Figure 2). Next, type ‘python get-pip.py’ into the command prompt and PIP will be installed. (Figure 2).

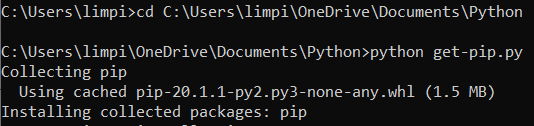


Figure 2: Installing PIP in the command prompt window

PIP can now be used to install the necessary packages by typing ‘pip install click’ and then ‘pip install pandas’. Following each command, the package will be installed. Lastly, import the packages by first entering ‘python’, this will show the Python program details acknowledging that you are now in the python interpreter (Figure 3), and then type ‘import click’ and then ‘import pandas’. You can now close the command prompt window

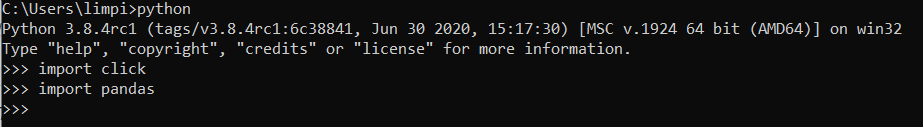


Figure 3: Importing packages for the external scripts used in the folders.

Viewing files

For those inexperienced with basic coding we recommend installing Notepad++ to view the command and batch files. This software is freely available at <https://notepad-plus-plus.org/downloads/>.

Example folders

Download and extract the ‘Example Folders’ zip file from [https://github.com/rcfgroup/gc-automation](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Frcfgroup%2Fgc-automation&data=02%7C01%7Cmjw77%40leicester.ac.uk%7C5755eaebcb7243596dfe08d81cf52b3a%7Caebecd6a31d44b0195ce8274afe853d9%7C0%7C0%7C637291186603018602&sdata=a9cxHXPVlS%2B%2BiNhMZ6NbXA4ENsflxu7rCcd80XN%2B1IU%3D&reserved=0).

Extract the zip file before using and ensure the ‘Example folders’ folder is extracted (or copied once extracted) to the ‘C:\temp’ location. If the ‘temp’ folder can not be located create a new folder.

The ‘Examples folder’ (now with a path at ‘C:\temp\Example folders’) contains three folders: 1 Match Template, 2 Export Match File and 3 Apply Match File. These three folders are examples to help a beginner become familiar with the concept of using the command line interface to integrate free and open-source software with commercial software for GC×GC data processing. The examples in the folders wouldn’t necessarily be used independently as part of a workflow but effectively demonstrate this new way of being able to process GCxGC data.

Please note, if using a different version of the commercial software (not v2.8r2), you may need to change the path for the Command Line. In the first and second example folders, this can be changed in the batch (.bat) file (Figure 4). In the third example folder, this can be changed in the python (.py) file (line 41, keeping the double slash formatting). The command line interface can be found in the program directory. An example path is:  
C:\GC Image\GC Image 2.8r2 GCxGC (64-bit)\bin\CommandLine.bat.

Right-click on the .bat file in the first or second example folder, or the .py file for the third example folder, and select ‘Edit with Notepad++’. In the .bat file, under ‘set GC\_IMG=’ paste the new path and then save the .bat file (as in Figure 4A). In .py file, under line 41 after ‘cmd =’ enter the new path using the double slash format shown in Figure 4B.

(A)

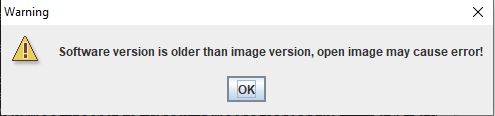


(B)



Figure 4: Changing the path for the command line interface in (A) the batch file and (B) in the python file.

When running the batch files (as described on page S-10) using a different version of the commercial software, the following message may appear.



Click ‘OK’ and the analysis will continue. The message will appear three times for each file being processed, click ‘OK’ to continue. This is not an issue for completing the examples, however, for custom workflows with more files the version information should be changed in the command and batch files.

**Explanation of example folders**

The first folder (1 Match Template) is a simple exercise, telling the commercial software to perform two tasks using the command line interface. These steps can also be performed in the GUI. This example is to get the user familiar with the translation between the GUI and command and batch files.

The second folder (2 Export Match File) tells the commercial software to perform to two simple tasks again (matching a template and exporting a summary report) but this time the batch file incorporates an external FOSS, in the form of a Python script. Once the commercial software completes the tasks, the Python script uses the outputs (template file and summary report) to automatically generate an exported match file. The generation of multiple match files is a task which can’t be batch processed within the GUI.

The third folder (3 Apply Match File) uses the processed chromatograms (Output folder) from the first folder (1 Match Template) and the exported match files (Exported Match Files folder) from the second folder, to perform a more complex list of commands, aligning the chromatograms using the Gros et al local alignment algorithm (available as plug-in).1 This time the commercial software is run through the Python script (as opposed to the Python script appending to the end). The Python script performs an automatic match between the batch number in the exported match files and the batch number in the chromatogram fiename. Once it has matched them, it uses the external algorithm automatically align each chromatogram using each batch’s unique match file. This intelligent iterative matching can not be performed within the GUI.

The example data in the first and third folder are chromatograms of a simple reference mixture with small variation in secondary retention times between the samples, especially for compounds with higher 2tR (e.g. aromatics). The files have three different batch numbers (180808 15, 180815 1, and 180815 9) representing samples run at different times The example data in the second folder are chromatograms of an n-alkane and aromatics mixture run on the corresponding dates as the reference mixture with the same batch numbers. To save memory space and make the demonstration as simple as possible the chromatograms are GC×GC-FID files, which have been exported a model image and saved as .gci files.

**Instructions**

1 Match Template folder - Instructions

In the ‘Input’ folder there are three chromatograms (six files in total, three .gci and three .bin files) labelled ‘ref mix’ with three different batch numbers (180808 15, 180815 1, and 180815 9). To run the analysis simply double-click the .bat file. The command line window will open showing each file being processed. The command window will then close and three processed chromatograms (six files in total, three .gci and three .bin files) will now have appeared in the Output folder. The number of chromatograms can be changed (e.g. >00s), and the processing time will change accordingly, however, this example highlights the simplicity of sharing and repeating a workflow and gets the user familiar with the interface. (see Results on page S-10).

2 Export Match File folder - Instructions

In the ‘Input’ folder there are three chromatograms (six files in total, three .gci and three .bin files). To run the analysis simply double-click the .bat file. The command line window will open showing each file being processed. The command window will then close and three processed chromatograms (six files in total, three .gci and three .bin files) will now have appeared in the Output folder. In the ‘Exported Match Files’ folder three .csv files will now have appeared. These are the exported match files for each batch based on the change in retention positions of the n-alkanes and aromatic compounds in the mixture. (see Results on pages S-10).

3 Apply Match File folder – Instructions

Firstly, copy the files of the three chromatograms of the reference mixture (.gci and .bin files) from the Output folder of the first example folder (1 Match Template), and paste them into the Input folder in this folder. Next, copy the three exported match files (.csv files) from the Exported Match Files folder of the second example folder (2 Export Match Files), and paste them into the Exported Match Files folder in this folder.

To run the analysis simply double-click the .bat file. The command line window will open showing each file being processed (Figure 5). The command window will then show a message saying ‘Process finished’ and ‘Press any button to continue’. Press the ‘enter’ button on the keyboard and the command line window will close. Three processed chromatograms (six files in total, three .gci and three .bin files) will now have appeared in the Output folder. In the ‘Exported CSV Files’ folder three .csv files will now have appeared. These are the chromatograms exported as a single-column vector .csv file. The processed chromatograms can be reviewed in Investigator (see Results section on pages S-11 – S-12).

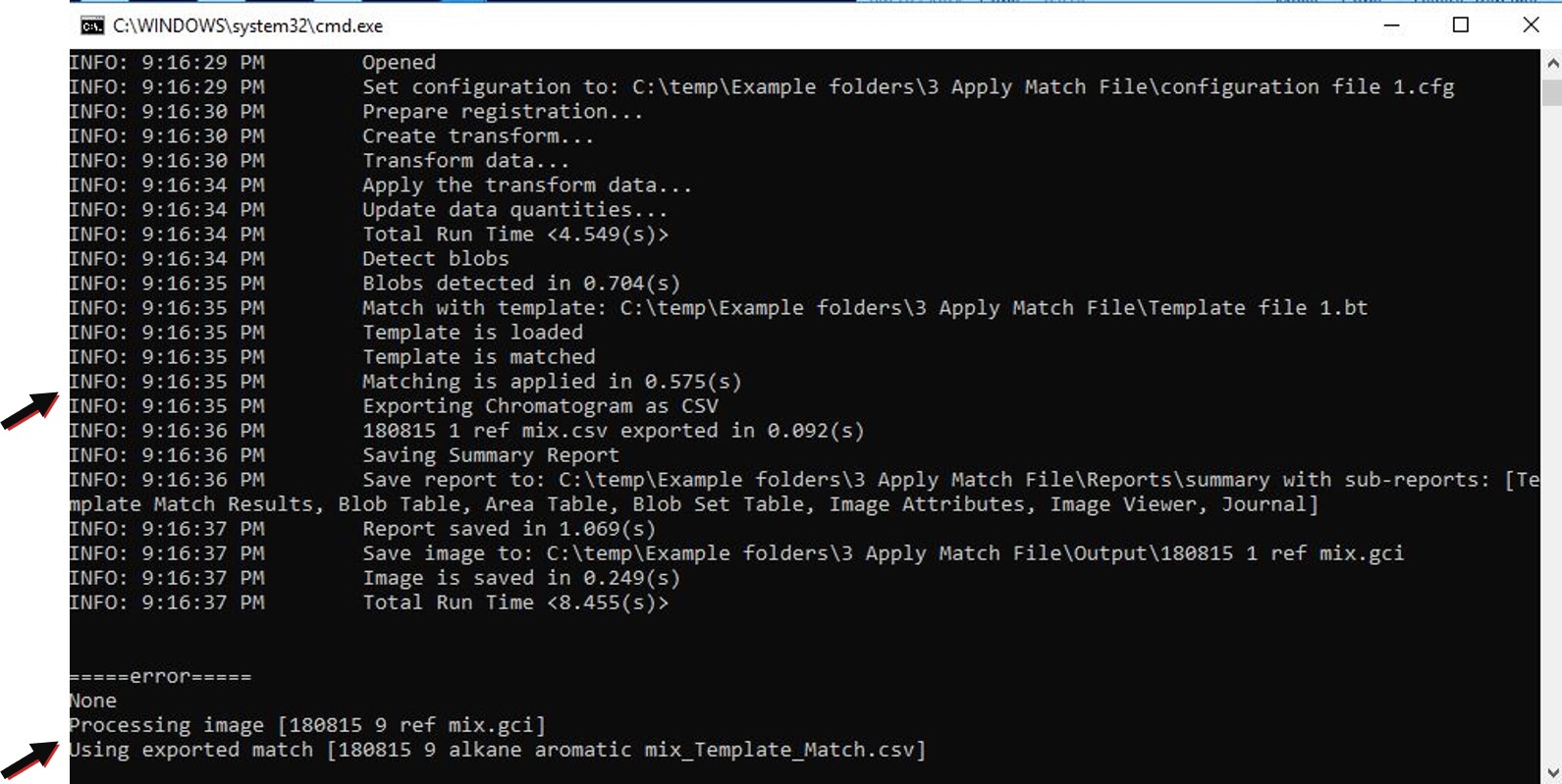


Figure 5: Example of command line window showing stream of processed files

**Results**

1 Match Template folder - Results

The Input and Output files can be viewed in GC Image. Open GC Image and click the yellow folder icon in the top left corner (Figure 6). Navigate to the C:\temp\Example folders\1 Match Template\**Input** folder, select a chromatogram to open. You will see a series of peaks, however, in the chromatograms from the Input folder the chromatograms will appear unprocessed without any peaks (or ‘blobs) detected. In the chromatograms from the Output folder the chromatograms will have the series of peaks detected (with yellow and red outlines around them) and the peaks above the hydrocarbon series with yellow outlines (the carbonyl, terpene and aromatic peaks) will have been matched to the template and have compounds names associated with them.

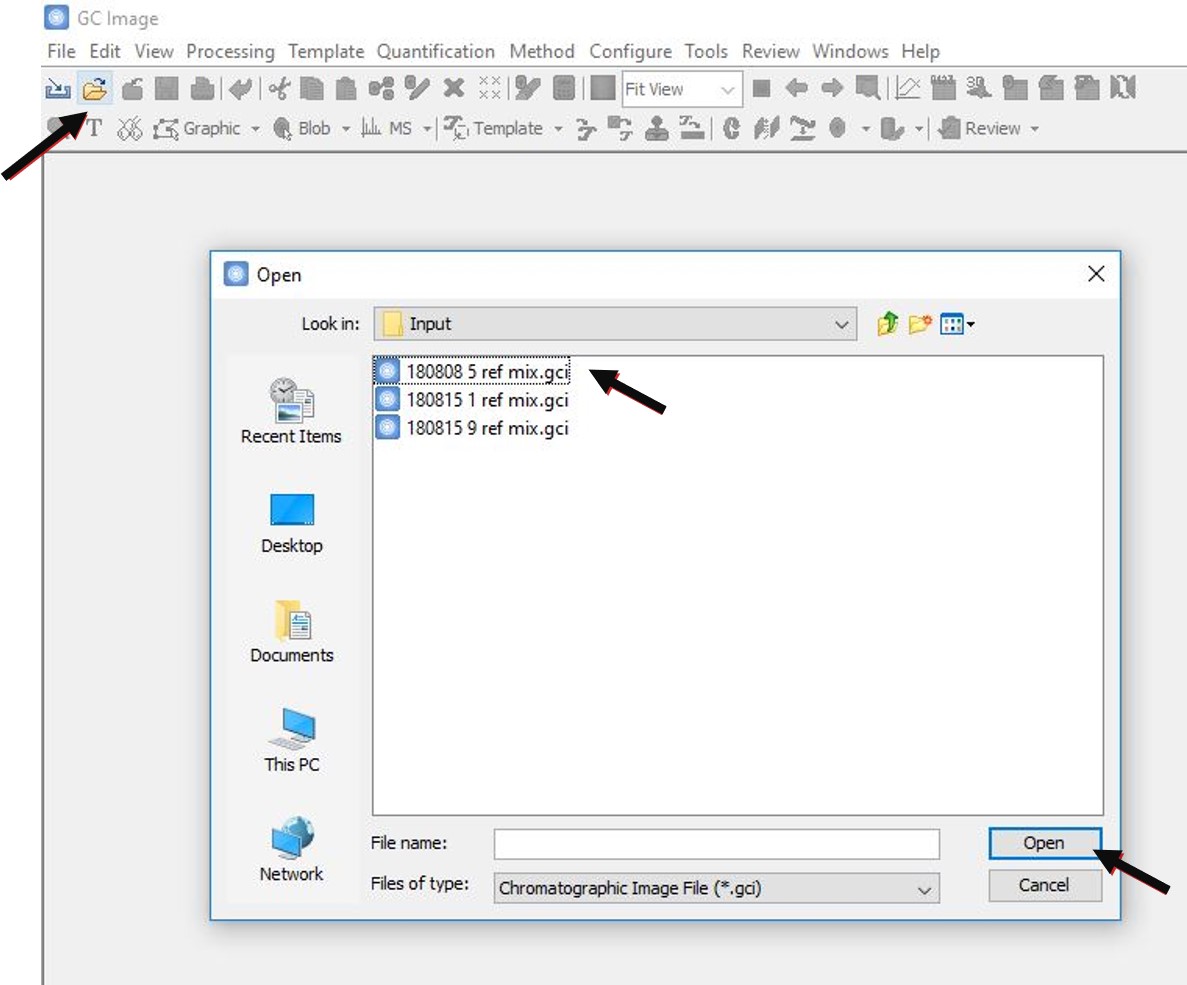


Figure 6: Opening an example chromatogram in GC Image

To access the processing steps that have been performed on the chromatograms, the user can open the command and batch files. To open the command file (.cmd) and the batch file, right-click on a file and select ‘Open with’ and then ‘Notepad’ or ‘Edit with Notepad++’.

2 Export Match File folder - Results

The chromatograms in the Input folder show the separation of a mixture of n-alkane and aromatic compounds separating across the 2D chromatographic space. The peaks were detected and a template applied labelling the peaks across all the samples. This can be done in the commercial software GUI or using the interface as demonstrated in the first example. The chromatograms in the Output folder are the same, only the alignment template, comprising a subset of the compounds, was applied to the peaks and a summary report of the peak positions generated as specified in the commands file. The Python script in the folder took the details of the reference positions of the subset of peaks from the template file, and the actual retention positions of the subset of peaks in the chromatogram from the summary report and combined them to reproduce match files (as can be produced one at a time in the GUI) for each of the chromatograms automatically. This example demonstrates how interfacing commercial software and free and open-source software can be used to automate custom iterative tasks.

3 Apply Match File folder - Results

The Input and Output files i.e. the chromatograms pre- and post-alignment using the Gros et al local alignment algorithm, can be compared in Investigator (part of the GC Image suite).

To do this, firstly open Investigator. In the top left of Investigator go to ‘File’ and click ‘Load Images’. Navigate to the C:\temp\Example folders\3 Apply Match Files\Input folder, select all three chromatograms holding the shift key and click ‘Open’.

In the ‘Load Options’ window (see Figure 7), *uncheck* the ‘Use configuration’ box and in the ‘Features’ tab select the ‘As is’ radial button. Next, click on the ‘Attributes’ tab, *select* ‘Analyse specific blob/area attributes’ and select the ‘Choose retention and response attributes’ radial button. In the list only *check* the ‘Retention I’, ‘Retention II’ and ‘Volume’ boxes, and uncheck the ‘Analyse specific blob set attributes’ box (Figure 7). In the ‘Class Assignment’ tab, select all three files by clicking whilst holding the shift key, click ‘Assign Class Label’ and choose ‘New Label’. In the box enter ‘Pre’ or an equivalent label and then click ‘Ok’, and then click ‘Ok’ again in the bottom left hand corner on the window. Investigator will then load the chromatograms into the program.

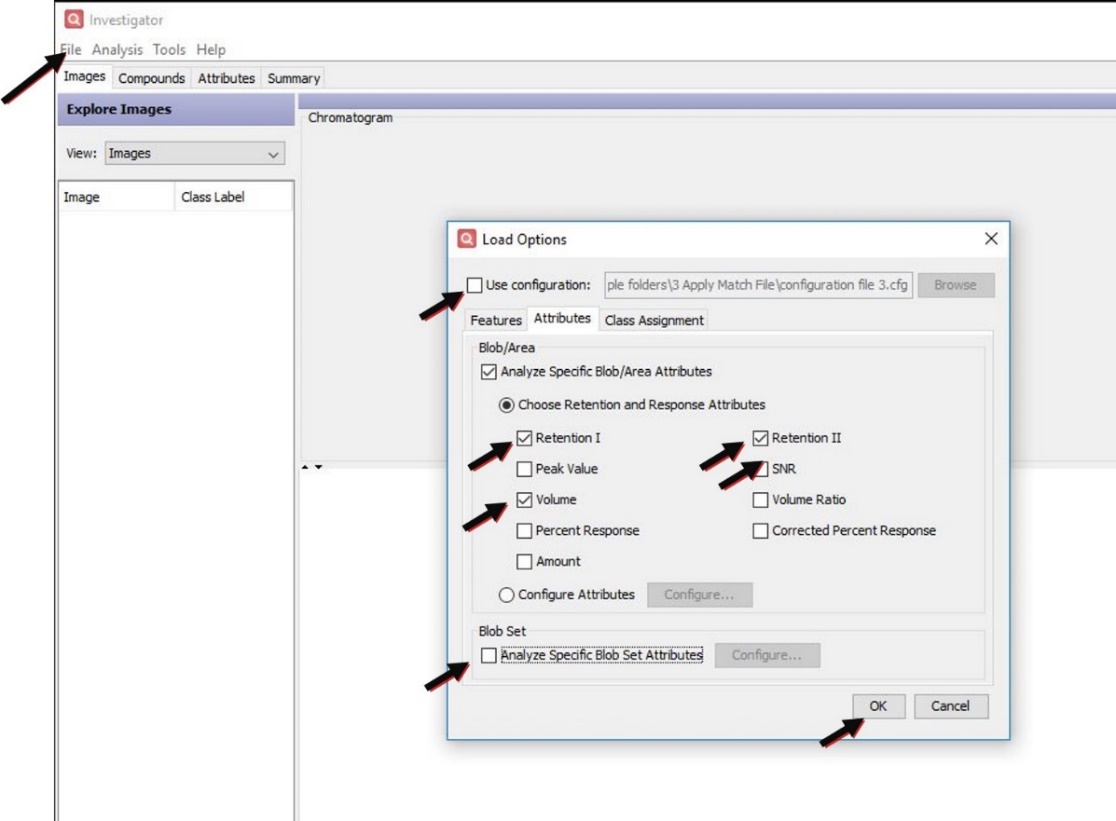


Figure 7: Loading the processed chromatograms in Investigator to compare the secondary retention times pre- and post-alignment

Next, repeat the above to load the processed chromatograms, this time navigating to the C:\temp\Example folders\3 Apply Match Files\Output folder, selecting all three chromatograms holding the shift key and click ‘Open’. All the options in the ‘Load Options’ window will now be greyed out, except for the ‘Class Assignment’ tab; highlight the new files and label them ‘Post’ or an equivalent label.

All six chromatograms should now be loaded into Investigator. If available in your version of the software, click ‘Analysis’ in the menu bar at the top and select ‘Custom settings’. In the ‘Custom Analysis Settings’ window, only check the ‘Class Mean’, ‘Class Stdev’, ‘Class %RSD’ and ‘Pairwise Mean Difference’ boxes and click ‘Ok’. If ‘Analysis’ is not available, go to the next step.

Next, click the ‘Attributes’ tab (see Figure 8) and in the left-hand column select ‘Retention II’. In the table on the right-hand side it is now possible to compare the %RSD(Pre) and %RSD(Post) columns. If ‘Analysis’ was not available to define the %RSD columns, the comparison can still be made by scrolling across to see the RSD columns labelled ‘Pre’ and ‘Post’. For all the compounds, the variation in secondary retention time is reduced in the post-alignment chromatograms (Figure 8). This analysis was implemented by double-clicking a .bat file. For further information on using Investigator please refer to the manufacturer guidance.

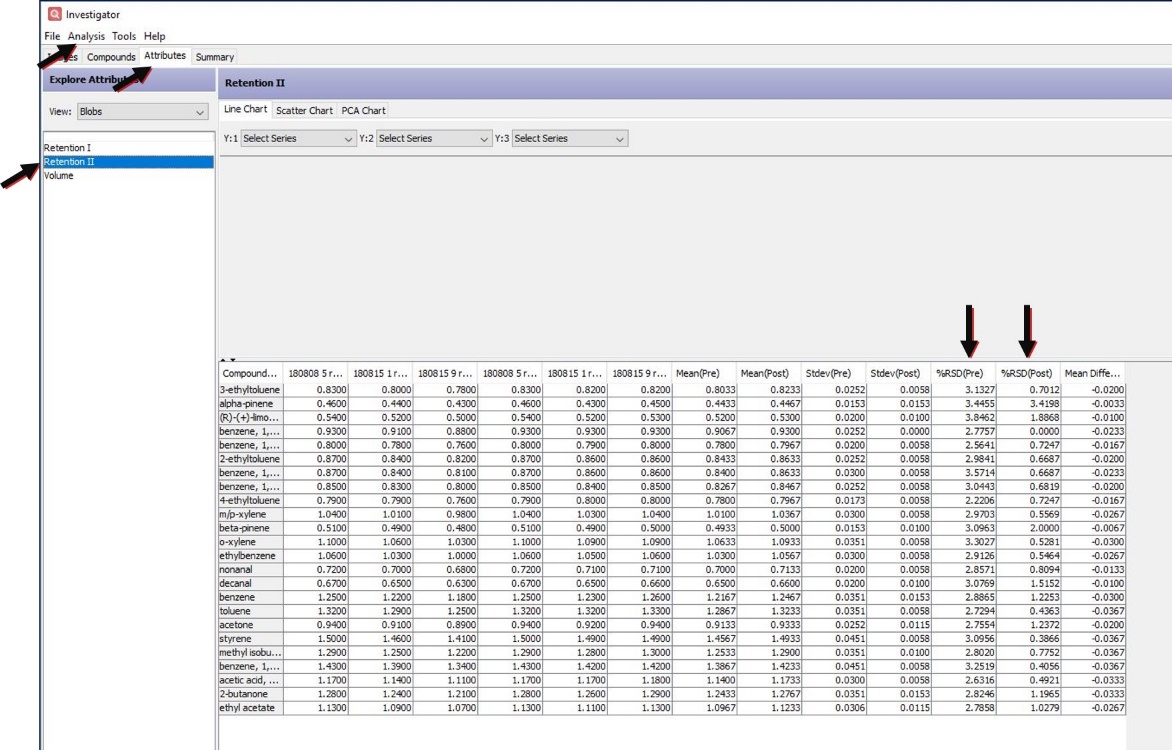


Figure 8: Comparing the variation in secondary retention time in Investigator, pre- and post-processing using the interface

In this example, the command file was run through the custom Python script, demonstrating how commercial and FOSS can be used to automate complex custom workflows. The external script instructed the software to find the date and tray number at the beginning of each chromatogram (e.g. a new function added by the user), and align the chromatogram (e.g. an existing function within the commercial software) using the algorithm developed by Gros et al (e.g. integrating published open-source code) based on the corresponding match files automatically generated (i.e. new custom function from previous example) from the n-alkane/aromatic reference sample with same date and tray number. In effect, alignment vectors were automatically exported and then used to align all the samples based on batch and tray number, correcting variation in retention positions captured by the n-alkane and aromatic reference mixture.

In this example, the reference mixture of terpenoids, aromatics and carbonyls represents samples, but for demonstration purposes using a reference mixture with known compounds (independent of the alignment mixture) helps demonstrate the workflow implemented through the interface.