Documentation for the Gros-Arey code
to align GC×GC chromatograms as implemented in Matlab

Version 1.1.3

Jonas Gros and J. Samuel Arey, EPFL, 2015.

Please cite the following article when publishing any results obtained by use of this software:

Gros, J.; Nabi, D.; Dimitriou-Christidis, P.; Rutler, R.; Arey, J. S. Robust algorithm for aligning two-dimensional chromatograms. *Anal. Chem.* **2012**, *84*, 9033‑9040.

1 The purpose of the algorithm

The Matlab code is designed to correct the small run-to-run shifts in retention times in GC×GC chromatograms. The code was also successfully applied to the correction of shifts resulting from moderate differences in GC×GC instrument programs (see Gros et al. 2012 for the details of the case studied).

2 The tested validity of the algorithm

The algorithm was tested for chromatograms resulting from comprehensive two-dimensional gas chromatography (GC×GC) coupled to a flame ionization detector (FID) and to a micro-electron capture detector (µECD), as reported in Gros et al. 2012. It is believed that the algorithm should be applicable to any two-dimensional separation coupled to a univariate detector. However, some careful evaluation is advisable for very different systems. (refer to Gros et al. 2012 for the systems tested)

3 What the Matlab code does

It takes a target chromatogram and aligns it to a reference chromatogram, thereby generating an aligned chromatogram. The retention times of peaks in the aligned chromatogram should be closer to the retention times of peaks in the reference chromatogram than is the case between the target and reference chromatograms.

4 Organization of the model file directory. Where to find what.

The model code is organized as follows. From the base directory two folders are present, called `users/`, and `model_code/`.

These two folder names should not be changed.

The user should only need to operate from within the folder called `users/`. Normally, nothing should be changed or adjusted in the `model_code/`.

Within the folder called `users/`, the organization of folders and files is user-defined. The user can define directory paths with the following two model variables in the file `main.m`:

A) `input_path`. This variable indicates the directory path location of the input files. The input path variable is set in the file called `main.m`, and it assumes that `main.m` is located in the directory `users/`. The `input_path` variable also assumes that the indicated directory exists.

B) `output_path`. This variable indicates the directory path location of the output files.

Note: both `input_path` and `output_path` should be relative paths, starting with `users/`; `users/` should be situated in the program base directory.

5 Steps for use of the algorithm

5.1 Prepare input files

The model requires the reference chromatogram, the target chromatogram, and the positions of the alignment points in both the target and reference chromatograms. The required structure of these files is described in turn below:

A) The reference and target chromatograms.
(variables `Reference_chromatogram_file` and `Target_chromatogram_file`)
By default, these chromatograms should be csv files containing one long column vector of signal intensity values, separated by a comma (,) or semi-colon (;) (as can be exported from GC Image). The length of the file is assumed a multiple of the product of the sampling rate multiplied by the modulation period, and any last additional values would be ignored. Other file types might be used (two-dimensional csv files in the GC Image format, or multi-column csv files exported from ChromaTOF, in which case the column labeled "S1" or "TIC" is imported).

B) The positions of the alignment points in the reference and target chromatograms.
(variables `Reference_alignment_pts_file` and `Target_alignment_pts_file`)
These csv files should contain two columns with first and second dimension retention positions of alignment points, respectively, in units of pixels. The files can either be generated with the Matlab file `select_alignment_points.m` or with the help of an external software. The convention retained here is that the first pixel in the chromatogram has the position (1,1) in pixel units. When importing peak positions from an external program, the user is advised to check for consistency.
Note: the order of the alignment points in the two files should correspond.

The positions of the alignment points can be selected within Matlab by running the file `select_alignment_points.m`, and following the displayed explanations. This file should first be edited to provide the code with the necessary information to import the chromatogram. Additional explanations regarding this functionality are provided in the Appendix (title number 8) at the end of this document.

How to select good alignment points?

- How to identify corresponding peaks? The user needs to be sure that the peaks selected correspond to the same analyte in the reference and target chromatograms. This can be ascertained using standards, but visual inspection is sufficient in many cases involving chromatograms sharing a high degree of similarity.
- How many? The number of alignment points required to achieve good alignment may vary for widely different chromatograms. In practice, we found that a minimum of ~10 alignment points is sufficient for the alignment.
- Where? We advise that alignment points are chosen in each part of the chromatogram that exhibit shifting trends different from neighboring regions, and that most of the part of the chromatogram that is of interest to the user is within the convex hull of the alignment points chosen.
- And finally, the user can also proceed in an iterative way: in case the result of the alignment is not satisfactory enough, the set of alignment points can be modified, and the alignment code run again.

What is the position of a peak?

The definition retained here is that the position of a peak corresponds to the position of the pixel of this peak that has the maximum signal intensity value.

The use of other definitions for the position of a peak is possible, but requires that positions of the peaks are determined outside of these Matlab codes (i.e. not with the help of `select_alignment_points.m`).

5.2 Adjust parameters in `main.m`

Adjust the parameter settings that appear in `main.m`. This file can be read and modified from within Matlab or using a generic text editor. This is the only Matlab file that you need to adjust, for normal use of the alignment code, unless you need to select alignment points using `select_alignment_points.m`. Most of these parameters are self-explanatory and/or discussed above.

One of these parameters is discussed below.

`Typical_peak_width`: this variable contains the first dimension (first element) and second dimension (second element) typical size of a peak, in units of pixels. It corresponds approximately to the number of pixels corresponding to two standard deviations of a peak (assumed Gaussian), for a typical peak. The determination of the typical peak width parameter is somewhat arbitrary, but the algorithm results are not very sensitive to this parameter. Therefore, a visual determination by inspection of one-dimensional slices of a typical peak (ideally close to the center of the chromatogram) should suffice.

6 Name and contents of the output file

The aligned chromatogram is saved as a one-dimensional column vector in a csv file. The name of the file is the same as the name of the target chromatogram file, with the string "_ALIGNED" appended.

7 Figures displayed

After the completion of the alignment, two figures are displayed:

A) A three-panel figure entitled "GC×GC chromatograms": the reference, the target, and the aligned chromatogram. Each of them is overlaid with the positions of the alignment points as black circles.

B) A two-panel figure entitled "Difference chromatograms": pixel-by-pixel difference chromatograms between the reference and target chromatogram (first panel), and between the reference and aligned chromatogram (second panel). Pixels that appear blue have a larger signal intensity value in the reference chromatogram compared to the other chromatogram. Pixels that appear red have a larger signal intensity value in the other chromatogram with respect to the reference chromatogram. Pixels that appear white have about the same signal intensity value in both chromatograms.
If the chromatograms are somehow normalized, especially for chromatograms sharing high compositional similarity, a better alignment should be highlighted by a whiter difference chromatogram (less red and blue regions).
(If your chromatograms correspond to samples having very different compositions or if they are not normalized, please consider only the "GC×GC chromatograms" figure.)
("Difference chromatograms" are presented in the article "*Tracking the weathering of an oil spill with comprehensive two-dimensional gas chromatography*" by R. K. Nelson et al., *Environmental Forensics*, 2006.)

8 Appendix: selecting alignment points within Matlab

8.1 Selecting alignment points manually (default)

The file `select_alignment_points.m` allows an interactive selection of alignment points within Matlab. The file should be edited to provide a chromatogram and basic information. The parameters are self-explanatory, and most of them are very similar to those in `main.m`.

When running the file `select_alignment_points.m`, explanations are displayed within Matlab main window, and a figure of the GC×GC chromatogram is displayed. Following the explanations displayed in the main window, the user can select a peak by clicking twice around it (the pixel with the maximum signal intensity value within this rectangle is selected as the position of the peak).

The positions of the alignment points are saved within a csv file that has the same name then the `chromatogram_file`, but with "_alignment_points" added to it.

<u>**Warning**</u>: For normal use, the variables `skip_alignment_point_selection` and `Look_other_chromatogram` should be set to 0. The next section explains what happens when one (or both) of these two variables is set to a value of 1.
<u>**Known bug**</u>: on some computers, there are display problems (the "crosshair" displayed by the Matlab function `ginput` used to select peaks remains visible also at the previous location when the cursor is displaced). Work around: on one (windows) computer, executing the command "`opengl software`" solved the issue.

8.2 Guessing the positions of alignment points in other chromatograms (optional)

<u>**Warning**</u>: guessing the position of an alignment point in one (second) chromatogram based on its position in another (first) chromatogram is solely based on taking the peak with the largest intensity value in the region of the second chromatogram close to the position of the peak in the first chromatogram. As an automatic tool, it can save time, but it can also lead to errors. <u>A careful check is strongly advised</u>!

Errors are principally to be expected if a peak is close to other higher peaks and/or if the extent of shifting is high. By design, this tool cannot work for the large retention time shifts generated by the use of different GC×GC instrument programs.

The user has two choices:

A) The user wants to select alignment points in a first chromatogram using Matlab, and then to check if their positions in one or several other chromatograms can be automatically found.

In this case the user needs to set the variable `skip_alignment_point_selection` to `0` (as you don't want to skip the selection of alignment points) and the variable `Look_other_chromatogram` to 1.

Below the variable `Look_other_chromatogram`, you need to provide the list of files corresponding to the chromatograms in which you want to look for the positions of the alignment points. This is done with the variable `other_chromatogram_file = {chromatogram1.csv', chromatogram2.csv'}`. You can provide any number of chromatograms.
The variable `window_size` sets the size of the window (first and second dimension, in pixels) for looking for a peak in a second chromatogram in a window around the position as reported from the first chromatogram. <u>The default ([2,50]) should work in most cases</u>, but editing might be required, especially regarding the second value, depending on acquisition rate. Larger values allow finding peaks further away from their position in the first chromatogram but will also tend to increase the possibility of wrong guess.
Note: the code makes sure that it found a peak position (defined as the position of the pixel of the peak with the maximum signal intensity value) by conducting an iterative search with a smaller window until the position obtained through this search is converged.
For each chromatogram, a figure is displayed and the user is asked (in Matlab main window) whether he wants to save the guessed positions of the alignment points; the naming of the saved file follow the same convention as explained above.

B) The user only wants to check if the positions of alignment points in one or several other chromatograms can be automatically guessed from their (already known) positions in a first chromatogram.

   In this case, the variables `skip_alignment_point_selection` and `Look_other_chromatogram` should be set to 1.

   The variable `alignment_points_to_guess_file` should contain the name of the file that contains the positions of the alignment points into the chromatogram defined by the variable `chromatogram_file`.
   The code will display the position of the alignment points in the original chromatogram, as black circle. Then it will iteratively go through the files in `other_chromatogram_file` and display the chromatogram overlaid with the guessed positions displayed as pink circles. In the main Matlab window, the user can answer whether he wants to save the results or not, depending if the guessed positions of the alignment points appear satisfactory on visual inspection.

Contacts:

For questions, problems, or bug reports, feel free to contact Jonas Gros (gros.jonas@gmail.com) or J. Samuel Arey (arey@alum.mit.edu).