Université Grenoble Alpes
Háskóla Íslands
Universitetet i Bergen

Baptiste Bordet (Baptiste.Bordet1@protonmail.com)

Documentation of Community Automatic Liquid Water Isotope Calibration-tool (CALWIC)

Last Update : 03/01/2024

Project in collaboration with :
Hans Christian Steen Larsen
Daniele Zannoni
Arny Sveinbjörnsdottir
Melanie Behrens

Contents

1 Introduction to CALWIC

CALWIC is a python program designed to process water isotopes measurements in liquid injection mode. CALWIC aims to provide an highly customisable tool while having high constraints on the user inputs to avoid as much as possible human errors. We aim this code to be suitable for a wide range of people no matter what is your knowledge on both water stable isotopes measurements or programming. We will start this documentation by a brief description of the effects occurring during measurements of water isotopes.

1.1 Memory effect

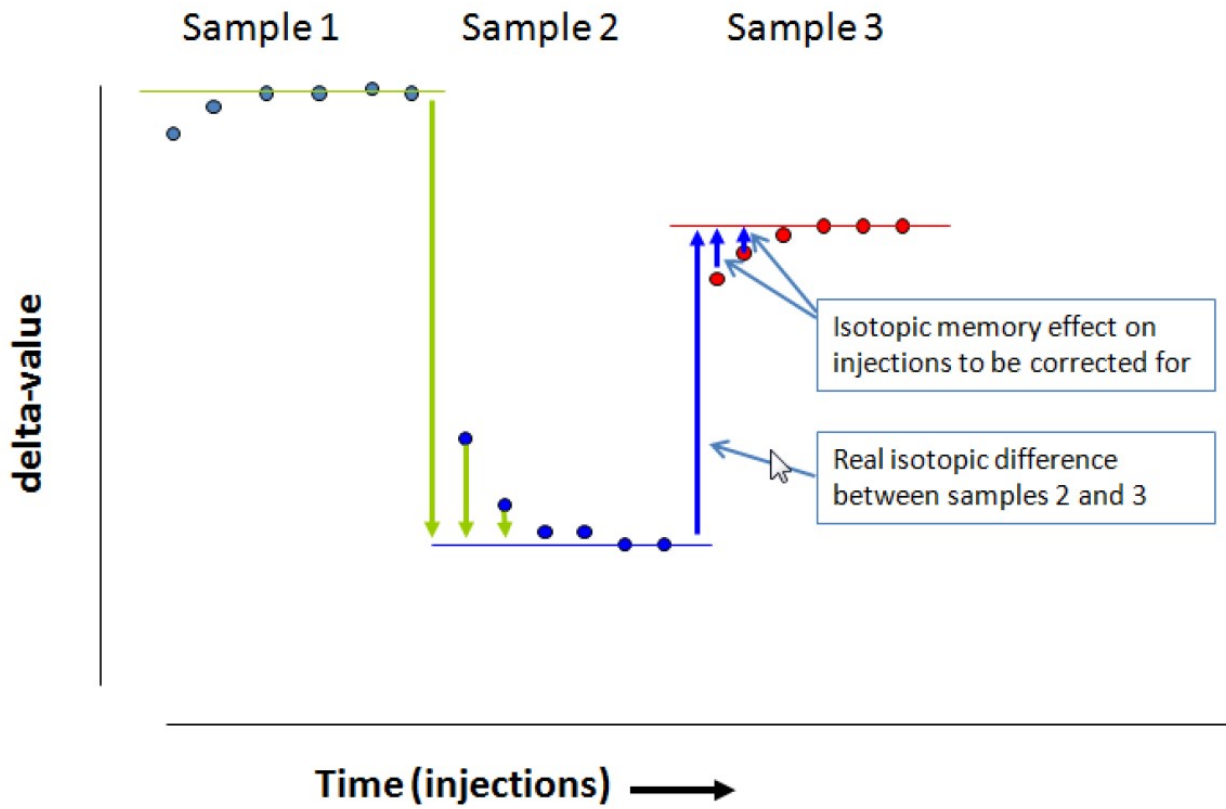

Figure 1: Illustration of the memory effect, taken from the SICalib user manual (Gröning [2018])

Since the injection system of water isotopes analyser is not perfectly hydrophobic some water will stick to the walls of the system (tubing, syringe mostly) during the injection of the sample. When injecting the next sample, some of this residual water will be blown away from the injection system and will be measured along with the new sample. This effect called memory effect produce results as shown in figure 1. The first injections of one sample are influenced by the value of the previous sample and this influence decreases with the number of injection. In CALWIC there are two methods implemented to correct this effect:

- The method suggested by van Geldern and Barth [2012] who proposed to evaluate the proportion of residual water in the mixing with standards at the beginning of a sequence.

- The method suggested by Gröning [2011] who proposed to evaluate the residual water through a two memory reservoir model.

1.1.1 Van Geldern correction method

To evaluate the proportion of residual water, a constrained non-linear optimisation is applied on standards' injections. This only means to adjust the memory coefficients (i.e. the proportion of water at each injection) to minimize the standard deviation of all the standards combined. The method used to calculate the memory coefficients is a constraint non linear optimization performed with the COBYLA algorythm (Constrained Optimization BY Linear Approximation). Details on the implementation can be found here : Scipy Community [a], Scipy Community [b].

From these coefficients the correction of all the values can be made through the formula given in van Geldern and Barth [2012] :

$$\delta^n_{j(mc)} = \delta^n_{j(raw)} + (1 - m_j) * (\delta^n_{j(raw)} - \delta^{n-1}_t) \tag{1}$$

where :

- $\delta^n_{j(mc)}$ is the memory corrected value of the j^{th} injection of the n^{th} sample,
- m_j is the memory coefficient of the j^{th} injection
- δ^{n-1}_j is the true isotopic value of the previous sample.

1.1.2 Gröning correction method

The method is subdivided into two steps :

- First, a single factor calculation computes an approximation using the last three injection to correct the current injection.
- Then, an exponential where the parameters are set manually is used to correct more accurately each injection.

Gröning's method is based on two successive correction. The first one assumes a single memory reservoir with "a constant relative memory decrease with time" (Gröning [2018]). The second one assumes two memory reservoir represented by two exponential laws combined to represent their interactions.

The first part is all done automatically. It is based on the evaluation of the memory contribution f for each pair of samples with the following formula :

$$f = \frac{a_j}{a_{j-1}} = \frac{true_j - meas_j}{true_j - true_{j-1}}$$

j refer to the sample number and starts at the second sample.

To take into account of the last three samples f is reduced :

$$f = f * (1 - f - f * f)$$

Then a mean of all f is done excluding the ones with two little difference in a_i (a_i should be more than 0.4 ‰ for $\delta^{18}O$, 4 ‰ for δD and 0.4 ‰ for $\delta^{17}O$) and the f negative (physically impossible).

Finally the first correction is applied with the following equation :

$$\delta_i^{first\ correction} = \delta_i^{raw} + f * (\delta_i^{raw} - \delta_{i-1}^{raw}) + f^2 * (\delta_i^{raw} - \delta_{i-2}^{raw}) + f^3 * (\delta_i^{raw} - \delta_{i-3}^{raw}) \quad (2)$$

Then to start the second correction, a percentage is calculated based on the following formula :

$$corr1 = \frac{\delta_i^{first\ correction} - true_j}{true_{j-1} - true_j} * 100$$

This percentage is then adjusted with the exponential parameters, alpha, beta and balance. These parameters are specific to each instruments.

$$corr = corr1 * (balance * e^{(-\alpha*(i-1))} + (1 - balance) * e^{(-\beta*(i-1))}$$

Finally this correction term is applied to the data :

$$\delta_{mc} = \delta_i^{first\ correction} - corr * (\frac{true_{j-1}}{true_j} * 100) \quad (3)$$

1.2 Calibration

The calibration is done by measuring standards with the instruments and comparing their values to the values measured by references.

The values are then corrected again with the standards. The memory corrected standards (all except the first one which is used as a rinsing one) can be used to create a calibration curve. The calibration curve is evaluated with the scipy *linregress* function (Scipy Community [c]).

2 How to install CALWIC ?

2.1 Download from github

In order to download CALWIC you can click here. This will download a zip file with all the modules. If you are familiar with github you can clone the project with the following command in your terminal:

```
git clone (repo_link) <local directory>
```

2.2 Install CALWIC with Anaconda

The first recommendation is to create a new environment to avoid any conflicts of the installed packages with your previous configuration. Open an Anaconda prompt. Use the following command to go to the unzipped folder of CALWIC :

```
cd path_to_the_folder
```

To create the new environment use the command below :

```
conda env create -f requirements.yml
```

Note : This command might take a few minutes to run since all the packages required for CALWIC will be downloaded and installed.

This will create an environment called CALWIC_env which can be activated with the command :

```
conda activate CALWIC_env
```

The environment has to be activated each time you want to run CALWIC. It can also be done in Anaconda Navigator by choosing it instead of base on top left of the window. After this you can open the file called CALWIC_main.py in Spyder and run it.

2.3 Install CALWIC with another Python distribution

You will need to install the packages included in the requirements.txt file. This file is included in the file you downloaded. The method to install it is up to you, depending on your IDE but the command to install all the required packages is :

```
pip install -r requirements.txt
```

Then you can run the CALWIC_main.py file and start processing. If you want to use the syringue tracker you will need to run first_launch.py first (see section 6.6 for more information.

3 How to make a run ?

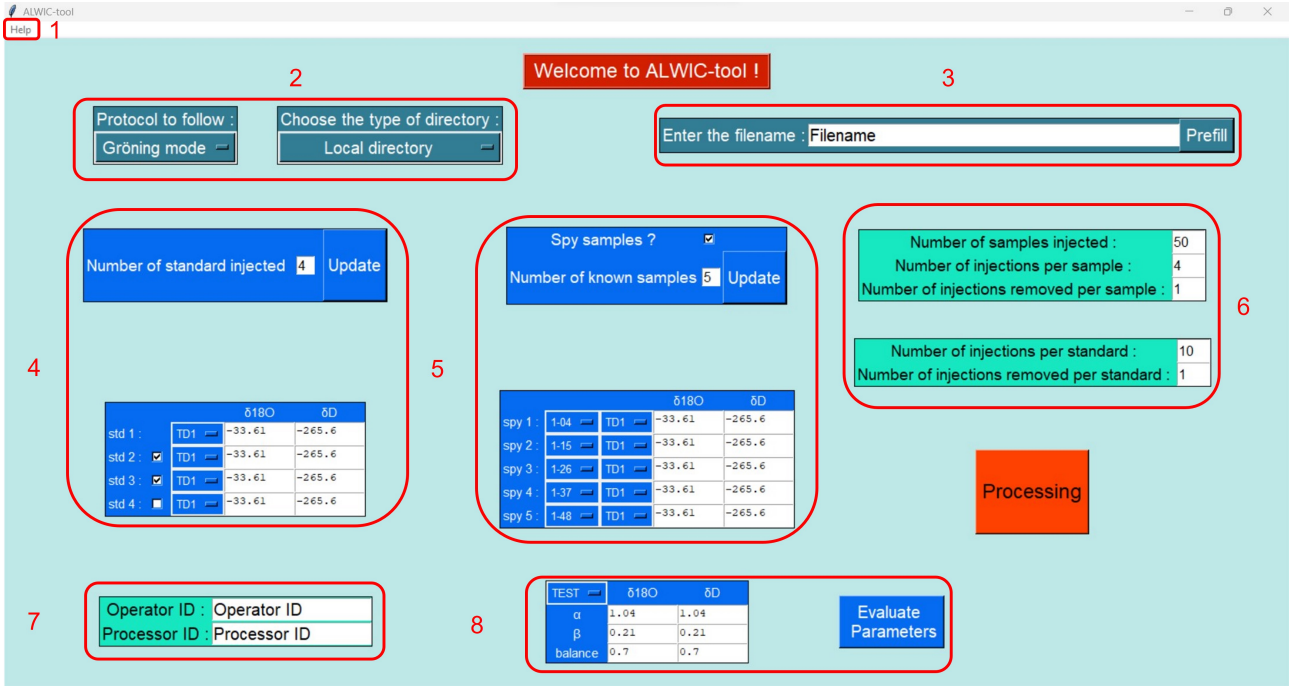

Figure 2: Screenshot of the interface of CALWIC

After starting a run a first page appears similar to the figure 2. There are different parts on this page numbered from 1 to 7 :

- 1 : Help links, they will redirect either directly to this document or to the github page of CALWIC.
- 2 : Here you will choose which method to correct the memory effect. You also have to choose if you want to correct the $\delta^{17}O$ as well. On the second box you need to chose from where you want to extract the file (see section 6.2 for how to set-up a drive account).
- 3 : Here you have to write the filename without extension. If you chose the local directory option on the previous point this will be already filled. Once you done this you can use the prefill function to prefill the points 4, 5 and almost all of 6. To be able to do the prefill the file must be customized before the measurements (more information on this on section 5.4).
- 4 : Here you first need to enter the number of standard injected at the beginning of the sequence with a maximum of 6 (This can be increased on the code, see section 6.7 to do it). Then click on the update button. A table will appear where you will be able to modify only two things : the tick-boxes and the names of standards. The tick-boxes control which standards are used for the calibration (on means the standard is used for calibration). The names of the standards are controlling the true values of these standards. The values are set in a different file before running CALWIC, see section 6.4 to know how to modify this file.
- 5 : Here is an optional part to fill. If you don't have any known samples in your run you can go to the next point. Else, you need to check the little tick-box and enter the number of known samples you have over the run with again a maximum of 6 known samples which can also be modified on the code, see section 6.7 for more information. Also if you're using a quality control standards you can include it here. Before clicking the update button make sure that the filename is filled because the update button will need your file to display the table. On this table you need to indicate the port where you placed these known samples. And in the same way as the last point you need to set-up the true values by changing the names. To modify the file controlling this you can look at section 6.4.
- 6 : Here you just have to enter a certain numbers of parameters of the run which are in order : The number of samples (with the known samples !!) injected on all the run, the number of injections per sample (must be constant along the run), the number of injections you want to remove to calculate the mean and standard deviation on each sample, the number of injection for each standard (must be the same for all standards) and the number of injections you want to remove to calculate the mean and standard deviation on each standard, see section 4 for more details on those calculations.
- 7 : Here is an optional part. You can indicate who ran the measurements and who is processing the data.
- 8 : This part is only shown when you choose one of the Gröning's method. You will be able to select your instrument's name. This will then modify the parameters used for the

exponential correction of Gröning. The values are set in a different file before running CALWIC, see section 6.5 to know how to modify this file. Additionally, you can define a new set of parameters by clicking on the button *Evaluate Parameters*. This will open a new window where you will be able to modify the parameters, look at the influence of these changes on the results and then save the set in the file previously mentioned. The Gröning's parameters are very stable over time and unique to each instrument, this is why only one file can be used to evaluate the parameters.

When all the steps have been made you can click on Processing. The memory correction process will start and show the first result page.

4 Output of CALWIC

If you choose the van Geldern method to do the memory correction, a temporary window will appear before doing the calculation. This window allows you to remove either one standard if you have multiple injections with issues or you can also remove one injection (but not more) of all the injections of standards. This tool should be used carefully and only in the case of really outliers. It should not be used to improve the memory correction !

To start the processing click on next page. The window will be closed and the corrections will be made.

4.1 New windows

When the processing button is pressed and all the calculation done, a new window appears. It looks like fig. 3. The parameters of the correction are shown on this page.

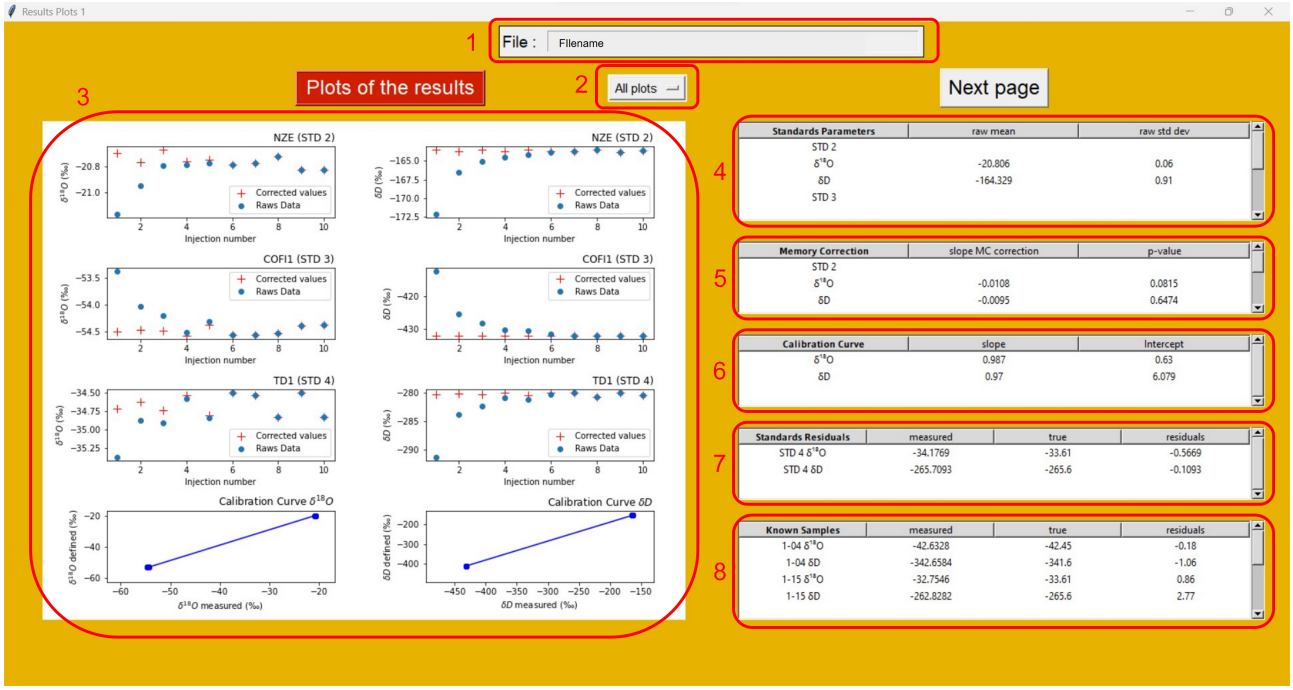

Figure 3: Screenshot of the first page of results of CALWIC

As previously this page is composed of different parts from 1 to 8 :

- 1 : Here it is a reminder of the filename. It may become really useful when you start to correct several files in a row.
- 2 : When this is set to all plots the page looks like the figure 3. But if you choose an other plot, the tables on the right will be overwritten to put a zoom on the plot you selected. To view again the tables, select all plots.
- 3 : You have here a summary of the correction. For each standard and isotope you will have one plot with the raw data and the memory corrected data (before calibration). These plots are auto-scaled in y-axis so be careful before trying to compare them. At the bottom you will have the calibration curves. These plots have the same scale for the two axis.
- 4 : Here you will find the mean and standard deviation of all the standards and isotopes. The mean and standard deviation are calculated on the values you choose not to reject on the previous page. These values are calculated on the raw values.
- 5 : This table will be different according to the memory correction method you use. If you used the van Geldern method, it shows the slope of the memory corrected values and its p-value. If you used the Gröning method, it will show you the mean single factor and alpha, beta and the balance term for each isotope.

  Note : Be very very careful with the p-values, it is calculated on a very limited number of values and p-value is a probability not a statistic (Wikipedia Contributors). If you are not aware enough of what are the limits do not use it !
- 6 : Here you will find the parameters of the calibration curves i.e. slope and intercept.
- 7 : This table is optional it will only show up if you don't use all the standards to show the residuals on the standard not used for calibration. The residuals are always defined by $residuals = measured - true$.
- 8 : Also an optional table. It will show up if you have known samples in your run. Again it will show the residuals of the known samples. The residuals are always defined by $residuals = measured - true$.

By clicking on next page you will open the second page of results (figure 4). Note that the first page of results remains open. On this new page the results shown are the samples values. Again there are several parts on this page from 1 to 4.

- 1 : Same as previous page, a reminder of the filename.
- 2 : This part shows several plots against the injection number.
- 3 : This table shows every mean sample (the mean being calculated by removing the number injection set at the beginning of the post-processing).
- 4 : This allows to save the data in different files (next section will describe it). You will be asked for the folder where you want to save the date. A pop-up window will appear after saving.

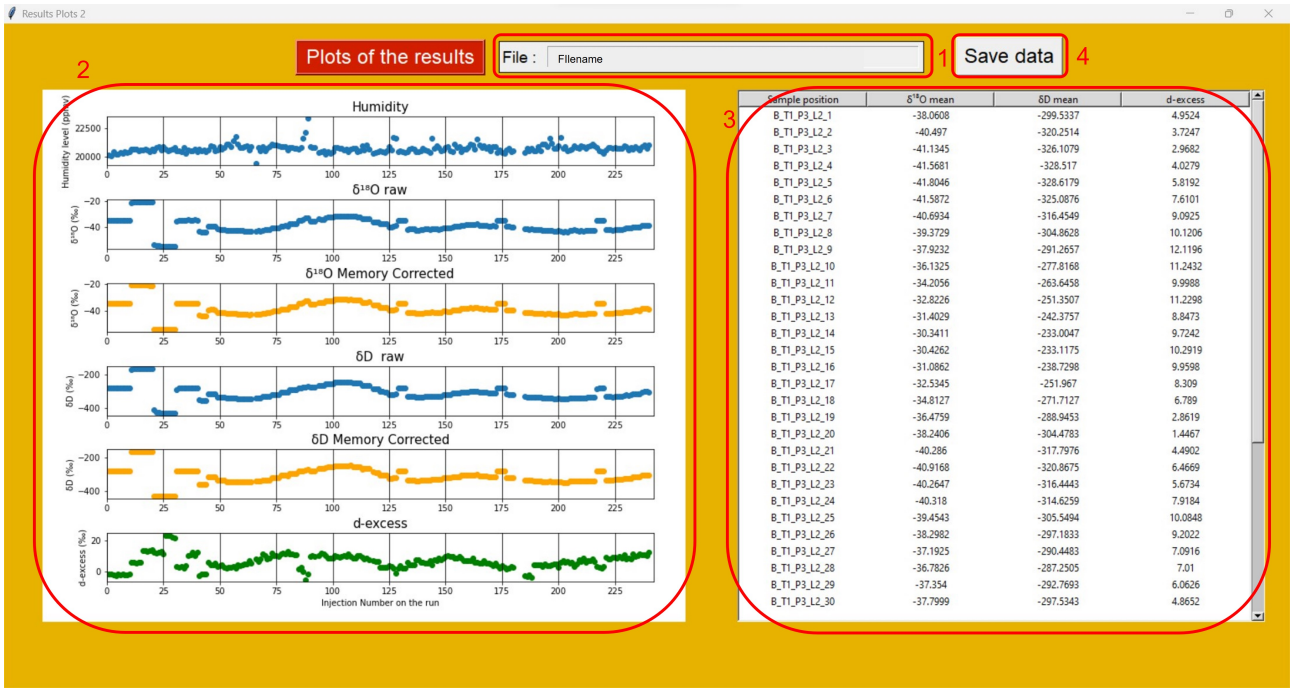

Figure 4: Screenshot of the second page of results of CALWIC

4.2 Saved Results

There are up to 4 files after each processing.

- 1. filename_final_file. It contains the input file with some columns added which are the memory corrected values as well as the calibrated values and some flags (more on it on section 6.1).

- 2. filename_std_parameters. It contains a lot of information about the run like starting and ending time, parameters of the correction, the standard deviation of the humidity (detrended) along the run, names of people who ran the file and processed the data, the correction method and the mean and standard deviation of all standards.

- 3. filename_spl_results. It contains the name of each sample as well as the mean and standard deviation of the last injections (the one you didn't choose to remove).

- 4. filename_control_samples_results and is optional. It is only saved when you have spy samples in your run. It contains the position of each control samples with their measured values, true values and residuals.

5 Options of treatment

CALWIC offers numerous options to process your data. They are described in the sections below.

5.1 Using google drive

Google drive can be used to store measured files and treated files. CALWIC has a module allowing it to communicate with google drive. It uses the google API (Application Programming Interface) which allows two machines to communicate in this case CALWIC and google drive. To do so the google account have to be set-up, it is described in section 6.2.

5.2 $\delta^{17}O$ mode

When measuring $\delta^{17}O$ this column is corrected and a new column for 17O-excess is calculated. The 17O-excess column is calculated with the following formula according to Luz and Barkan [2005]:

$$\Delta^{17}O = \ln(\delta^{17}O + 1) - 0.528 * \ln(\delta^{18}O + 1) \quad (4)$$

5.3 Control samples

In order to look at the short term drift and to evaluate the performance of the run, "spy samples" can be added to the run. They are standards measured as samples (with the same number of injection and corrected in the same way as samples). It is then possible to evaluate the residuals along the run and the change of these residuals to identify potential issues.

5.4 Prefill

This option is only available for Picarro's instruments.
CALWIC has a function to evaluate parameters directly from the file and to prefill it for the user. The filename needs to be filled and the following parameters are evaluated when it is possible : number of standards, True values of the standards, injection per standards (must be constant along one run), number of samples, injection per sample (must be constant along one run), if there are spy samples, if yes the port of each spy sample, the values of the spy samples.

Note : in order to prefill the true values of standards and spy sample the names of standards in the measurements files have to match with one standard name in the std_values.csv file (see section 6.4).

The set-up of this function is described in section 6.3.

5.5 Config_file

There is a configuration file used to set-up default values for data processing. It is located in the folder called "Files". It contains different parts :

- Parameters to be set by default : standard number, injection per standard, removed injection per standard, is there spy samples, spy sample number, sample number, injection per sample, removed injection per sample, operator id, processor id, instrument type (for future use), instrument identifier, memory correction protocol, type directory input files.

- Paths and extension : directory saving figures, directory saving files, directory input files, extension output files, extension figures, csv separator.

- Batch processing : if the files have exactly the same run architectures, the batch processing can be activated with the following parameters : is batch processing, batch processing mode, directory saving figures, directory saving files, directory input files. Batch processing is further described in section 5.6.

The parameter raw value if not corrected is used to set behavior of CALWIC in case the injection can not be corrected (the behavior is described in the last flag of the section 6.1).

5.6 Batch Processing

It is possible with CALWIC to treat multiple files which have the same characteristics (same length, same position of standards and samples, same standards injected). As said in the previous section the batch processing can only be activated in the config file with the is_batch_processing option which should be set to True. There are two modes of batch processing :

- Automatic : everything is done in the background nothing is shown to the user but all the figures are automatically saved. this mode requires the paths to files to be filled contrary to the next method.
- Manual : In manual mode the results are shown to the user who can reject the results and not save them. At the end of the processing of each file all the results page are deleted and the processing of the next file starts.

6 Miscellaneous

6.1 Flags

There are three flags in total. One is saved in the spl_results file. It indicates a standard deviation of each sample higher than the threshold (By default, the thresholds are fixed to 0.06‰ for $\delta^{18}O$, 0.21‰ for δD and 0.1‰ for $\delta^{17}O$. See section 6.7 to know how to change these values).

The two other flags are saved in the final_file. One is for the humidity level, if the humidity for one injection higher than 23000 ppmv or lower than 17000 a flag will be raised.

The other flag indicates if an injection has been corrected or not. There are three possible values : 0, the value hasn't been corrected either because it is the first standard either because there is NaN somewhere. 1, the value has been corrected. 2, the value has not been corrected and the final value has been replaced by the raw value. This has been set-up for the case where the last injection of a sample is missing the next sample can't be corrected by any method. Instead of losing the value of this sample the raw values are used and the user is informed of this with a pop-up window when the second page of results is opened.

6.2 Set-up a google account to interact with CALWIC

Setting-up the link between CALWIC and a google account is a tedious task but once it is done, it can be a very powerful tool to work remotely or with co-workers.

The first thing to do is to create a google account. It is probably a good idea to use an account only dedicated to this task since the account will be set-up in a developer mode.

In google drive two folders have to be created : one for the raw files named raws_data (the name does not matter since the search from CALWIC will look into the entire drive for the matching filename), one for the treated files named Treated_data (the name can be changed on the code, see section 6.7). The files can be anywhere in the drive, the only requirements is to not have other folders with the same name.

Now, the google account will be set-up to use the google API. Before starting this, be sure to be logged in with the account which will be set-up. If you have a different default account the solution is to log-out and then reconnect with the account you want to set-up in first. This will automatically change the default account.

The first thing to do is to enable the API. This can be done on the following link : `https://developers.google.com/drive/api/v3/quickstart/python`. When scrolling down you will see something similar to the figure 5. Click on Enable the API.

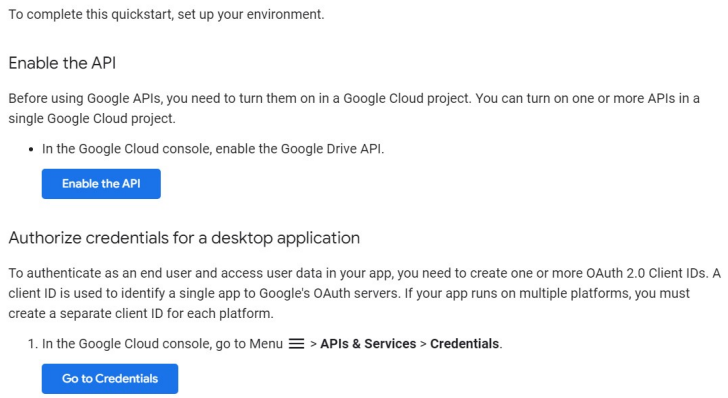

Figure 5: Box to click on to start the set-up

This will open a new window which is your google cloud console. The first thing to do is to create a new project. You choose a name and then enable the API. Then you can go back to the cloud console by clicking on Google cloud on top left of your window. Then on your left you have a panel. You will have to select the credentials in API and services as shown in figure 6.

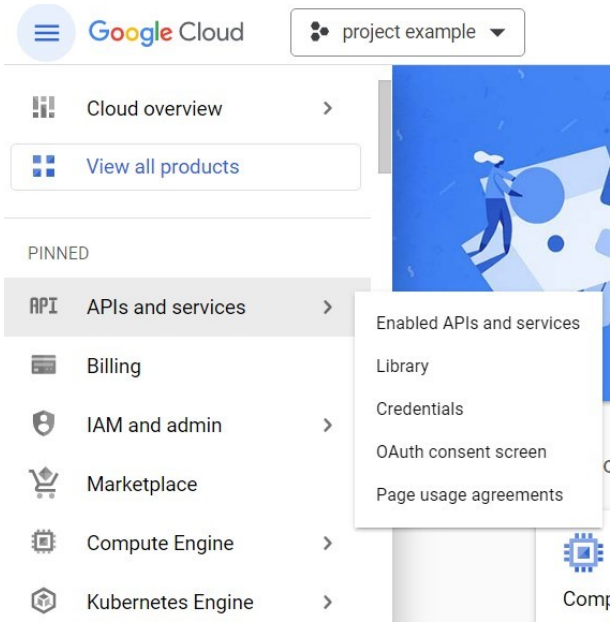

Figure 6: left panel where to find the credential control page

Then in order to create the credentials, you will have to click on create credentials and chose the API key. Then wait and close the pop up window. Then click again on create credentials and this time chose the OAuth ID. Then click on configure consent screen. Chose the user type Extern. Then, fill the App name, the user support name and the developer contact information which is your google email adress. Skip the scope page. On the next page there are the test users. You can fill up to 100 google account. Here you will fill every email which will use CALWIC. This can of course be modified later. And then you can ho back to the dashboard. And click again on create credentials and chose again OAuth ID. Chose Desktop App as application type and then create. When the pop-up window appears download the .json file and rename it client_code.json. You will need to move file into the app folder in the same folder as the python files.

6.3 Set-up the prefill function

In order to use the prefill function, the sample descriptor file has to be set up before the measurements. If you already use it to mark your samples name you only need to add one more column with title Identifier 2 and for vial indicate if it is a standard by putting STD, a sample by putting SAMPLE or control sample by putting SPY. The code is sensitive to upper-case be careful.

If you never used the sample descriptor, there is a file example in the example_files folder you just need to adjust it to your run.

When the Identifier 2 column is filled with STD, SAMPLE and SPY the prefill function is fully functional.

6.4 Standards true values

The standards values are stored in a separate file which must be closed when the code is running. The file is located in the File folder in the project folder. It is named std_values.csv.

There are five columns in the file, the first one is the description and can be used to add some annotation on the standards. The second column is the short name, this is the name which will be displayed on screen so be careful to have meaningful names in this column. The last three columns are the values for each isotopes expressed in ‰. The file can be saved with comma or semicolon separator without distinction but the function save as should not be used. It might lead to errors in the opening of the file by python.

6.5 Exponential parameters

The exponential parameters are stored in a separate file which must be closed when the code is running. The file is located in the File folder in the project folder. It is named groning_exp_parameters.csv. You can use the function in CALWIC to evaluate your sets of parameters but you can also import manually your parameters. In this case here are a few tips on how the file is built. The file is ordered by line. Each line is an instrument and all the parameters are ordered in different columns. The file can be saved with comma or semicolon separator without distinction but the function *save as* should not be used. It might lead to errors in the opening of the file by Python.

6.6 Trackers

To set-up the tracking you will need to run the file called first_launch.py. This will create a folder outside of the CALWIC working directory called CALWIC files : this folder contains to two types of a files : A file which save the name of all processed files and can be used to warn if the file has already been treated on the computer. The other type of file is a syringe tracker. This is little bit more difficult to use because it needs every file to be treated in the order it has been measured so the syringe changes reported in the file (which must be done manually) can be consistent with the situation of the instrument. It can be good practice to use this function with a different computer.

6.7 Changes in the code

- How to change the default values for the flag in sample result ?

  In the file save_results.py in the function save_sample_results_file you can change the thresholds.

- How to change the Treated_data name for google drive ?

  In the file called google_api in the function upload_files there is a parameters query which is set by default to Treated_data this can be changed.

Any other changes in the code can cause some unexpected side-effects. You can contact me via github (by opening an issue : (https://docs.github.com/en/issues/tracking-your-work-with-issues/creating-an-issue) or by creating a pull request if you want to add a feature or fix something in the code (https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request). It is also possible to send me an email at the following adress : Baptiste.Bordet1@protonmail.com. However since I am not working full time on it, I may take a few days to answer. If I did not answer your email after 2 weeks, please send another one.

7 Future developments

7.1 Other instruments

For now, only Picarro's files can be treated with CALWIC but in the future it will be expanded to other instruments which are doing measurements of water isotopes in liquid injection mode. Anyone who wants to help implementing other correction methods can send me an email at Baptiste.Bordet1@protonmail.com

7.2 Report

A pdf or html document to summarize the result of each processing could be considered but for now it is too much time consuming to develop, I will think about it only if people ask for it.

8 Licence

9 Acknowledgment

Thanks to Hans Christian Steen Larsen who initiated the project; Daniele Zannoni for all the support during the development of this tool; Arny Sveinbjörnsdottir for the comments made from a user perspective. Thanks to Manfred Gröning for the help provided in the implementation of his method. Thanks to Melanie Behrens for providing files for the tests. Finally a special thanks to Pauline Gayrin and Clara Berneuil for the emotional support as well as usefull discussion.

10 References

Manfred Gröning. *SICalib User Manual (Stable Isotope Calibration for routine δ-scale measurements)*. Terrestrial Environment Laboratory, International Atomic Energy Agency, Vienna, Austria, 2.16j edition, May 2018. `https://nucleus.iaea.org/sites/ReferenceMaterials/Shared%20Documents/Publications/TechnicalNotes/TELTechNote01_SICalibUserManual.pdf`.

Robert van Geldern and Johannes A.C. Barth. Optimization of instrument setup and post-run corrections for oxygen and hydrogen stable isotope measurements of water by isotope ratio infrared spectroscopy (iris). *Limnology and Oceanography: Methods*, 10(12):1024–1036, 2012. doi: https://doi.org/10.4319/lom.2012.10.1024. URL `https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lom.2012.10.1024`.

Manfred Gröning. Improved water δ2h and δ18o calibration and calculation of measurement uncertainty using a simple software tool. *Rapid Communications in Mass Spectrometry*, 25(19):2711–2720, 2011. doi: https://doi.org/10.1002/rcm.5074. URL `https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/rcm.5074`.

Scipy Community. Scipy minimize tutorials. `https://docs.scipy.org/doc//scipy/tutorial/optimize.html#constrained-minimization-of-multivariate-scalar-functions-minim` a. Accessed on 15/12/2022.

Scipy Community. Scipy minimize documentation. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html?highlight=minimize`, b. Accessed on 15/12/2022.

Scipy Community. Scipy linear regression documentation. `https://docs.scipy.org/doc/scipy-1.9.3/reference/generated/scipy.stats.linregress.html#scipy.stats.linregress`, c. Accessed on 15/12/2022.

Wikipedia Contributors. p-value. `https://en.wikipedia.org/wiki/P-value`. Accessed on 31/01/2023.

Boaz Luz and Eugeni Barkan. The isotopic ratios 17o/16o and 18o/16o in molecular oxygen and their significance in biogeochemistry. *Geochimica et Cosmochimica Acta*, 69(5):1099–1110, 2005. ISSN 0016-7037. doi: https://doi.org/10.1016/j.gca.2004.09.001. URL `https://www.sciencedirect.com/science/article/pii/S001670370400660X`.