

---

SpecXY  
Documentation & User Guide



---

**Nils B. Gies**

---

7 th of February 2024

# Contents

|       |  |    |
|-------|--|----|
| 1     | SpecXY . . . . .                           | 1  |
| 1.1   | Installation . . . . .                     | 1  |
| 1.2   | SpecXY for advanced MATLAB users . . . . . | 1  |
| 1.3   | SpecXY input data . . . . .                | 1  |
| 1.4   | SpecXY GUI . . . . .                       | 2  |
| 2     | SpecMaps . . . . .                         | 3  |
| 2.1   | SpecMaps GUI . . . . .                     | 3  |
| 2.2   | SpecMaps submodules . . . . .              | 5  |
| 2.2.1 | ClassificationModule . . . . .             | 5  |
| 2.2.2 | ImportMatchModule . . . . .                | 5  |
| 2.2.3 | CorrelationModule . . . . .                | 5  |
| 2.2.4 | MapCalculatorModule . . . . .              | 6  |
| 2.2.5 | ThicknessCalibrationModule . . . . .       | 7  |
| 2.2.6 | ExportModule . . . . .                     | 8  |
| 3     | SpecDB . . . . .                           | 9  |
| 3.1   | SpecDB GUI . . . . .                       | 9  |
| 3.2   | SpecDB submodules . . . . .                | 10 |
| 3.2.1 | SpotVisModule . . . . .                    | 10 |
| 3.2.2 | SimpleDeconModule . . . . .                | 10 |
| 3.2.3 | ExtractionModule . . . . .                 | 11 |
| 3.2.4 | SpecVisModule . . . . .                    | 12 |
| 4     | GeoCPlot . . . . .                         | 13 |
| 5     | Quick Start Guides . . . . .               | 14 |
| 5.1   | SpecDB . . . . .                           | 14 |
| 5.2   | SpecMaps . . . . .                         | 14 |
| 6     | SpecXY API . . . . .                       | 18 |
| 7     | SpecXY Data Structure . . . . .            | 18 |
| 7.1   | SpecXY Public . . . . .                    | 18 |
| 7.2   | SpecDB Public Functions . . . . .          | 18 |
| 7.3   | SpecMaps Public Functions . . . . .        | 19 |
| 7.4   | Public SpecDB Properties . . . . .         | 20 |
| 7.5   | Public SpecMaps Properties . . . . .       | 20 |

# 1 SpecXY

SpecXY is a user-friendly software solution for preparing, editing, extracting and comparing (spatially resolved) spectral datasets. SpecXY consists of three main modules for processing spectroscopic data. (1) The first module, SpecMaps, is designed for the investigation of spatially resolved spectroscopic data and allows various maps to be generated. This module provides a quick and efficient way to compare signals between different areas and extract average spectra for further calculations. It can be used to visualise and classify spectra, perform peak deconvolution from pixel spectra and to correlate spectral data with chemical data (e.g., from EMPA, LA-ICP-MS) or other numerical data of the same area. SpecMaps can handle any data type containing a map of wavenumber, energy, frequency, wavelength and intensity. (2) The second module, SpecDB, is designed as a spectra database with the ability to perform manipulations, advanced peak deconvolution, calculations (e.g., H<sub>2</sub>O quantification), match unknown spectra or visualise and compare important features in selected spectra. (3) GeoCPlot allows to classify data and helps to visualize data in correlation matrices.

## 1.1 Installation

SpecXY is available on Windows and MacOS (Intel) as standalone and does not require a MATLAB licence, but you will have to install the MATLAB Runtime libraries during the installation. For Mac users: It is recommended to start the installation with right click - open (Some versions of MacOS block the installation when starting it with a double click). The installation files including all needed libraries which are needed for computers without access to the internet are available on request. To install SpecXY execute the installation file and follow the instructions:



## 1.2 SpecXY for advanced MATLAB users

Execution in this method is only suitable for advanced users, requires a MATLAB licence and is not recommended for standard use of SpecXY. To run SpecXY without installation in matlab, the directory with the files must be added to the MATLAB path with the following syntax:

```
addpath(genpath('Your SpecXY directory'))
```

After that, modifications or extensions can be implemented using the MATLAB App Designer. Further information on execution, public properties and public functions is available in the SpecXY API.

## 1.3 SpecXY input data

SpecXY requires numerical data where  $x_{\text{signal}}$  can be wavenumber, wavelength, or energy, and  $y_{\text{signal}}$  can be intensity, absorbance, or counts (Fig. 2). The input file can be either a CSV file, a MAT file with the input matrix named AB, a single Spectrum BRUKER OPUS binary file, a PerkinElmer block structured file or an IMG file containing a hypercube. In the input matrix, the first column must contain the values for  $x_{\text{signal}}$ . The following columns must contain a single spectrum  $y_{\text{signal}}$  with the same number of data points as  $x_{\text{signal}}$ . Spatially resolved spectroscopic data can be either gridded map data, or spot measurements including a metadata file containing the *PixelPosition* expressed as  $X_{\text{position}}$  and  $Y_{\text{position}}$  coordinates for each pixel. In the case of gridded map data required for SpecMaps, the number of columns containing  $y_{\text{signals}}$  shall be equal to the product of  $nX \text{ Pixel}$  and  $nY \text{ Pixel}$ . If not provided by the data input, the user

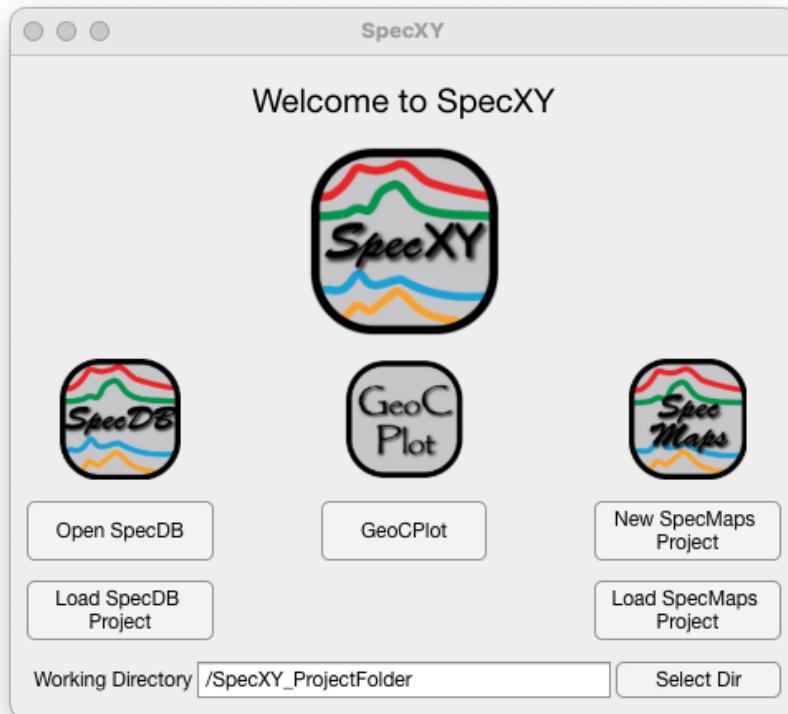
can select the values for  $nX\ Pixel$  and  $nY\ Pixel$  from a list of all possible product combinations resulting of the number of input columns containing  $y_{signals}$ . The first  $y_{signal}$  with the  $id=1$  is represented by the pixel located in the bottom left corner of the map. The pixel ids increase from left to right and from bottom to top. Compatibility with new data types and instrument types can be achieved by changing the data format or by a small adjustment to the import function, which converts the spectroscopic and metadata into the data structure described above. In addition to the required data input, a set of metadata can be imported and stored in the data structure. For example, to relate the spectral data to optical images or other numerical data, *MatrixLimits* mark the coordinates of the furthest extent of each dataset containing the minimum and maximum of  $X_{position}$  and  $Y_{position}$ . Additional information, that can be expressed numerically or as a string, can also be imported, by automatically matching identical strings of selected columns from the metadata table and the database table. The imported and matched data is then stored in the SpecXY data structure. SpecXY is capable of automatically importing different information file name or a metadata file. The following strings are supported at the moment and added to the database:

- Mineral: Common Mineral Names (e.g., \_CPX\_, \_Qtz, \_Ol\_, \_Garnet\_,...)
- Aperture:"\_XXap\_" (e.g., 25ap, 150ap...)
- Polarization: polmin, polmax, unpol, polround , "\_XXpol\_" (e.g., \_00pol\_, \_90pol\_)
- Orientation: (e.g., "\_alpha\_" "beta\_" "gamma\_" "IIC\_" "TC\_")
- Scans: "\_X...Xsc\_" (e.g., \_4sc\_, \_64sc\_)
- Thickness in micron:"\_deltaT\_XX\_", (e.g., \_deltaT\_, \_258\_)
- FTIR FPA binning: "\_nabin" "\_binXX" (e.g., \_bin8\_)

If your instrument or datatype is not supported yet, you can send an example of your data to [nils.gies@unibe.ch](mailto:nils.gies@unibe.ch).

## 1.4 SpecXY GUI

The graphical user interface (GUI) of SpecXY allows the user to set the working directory, open GeoCPlot, SpecMaps or SpecDB and load existing SpecMaps projects and SpecDB databases.

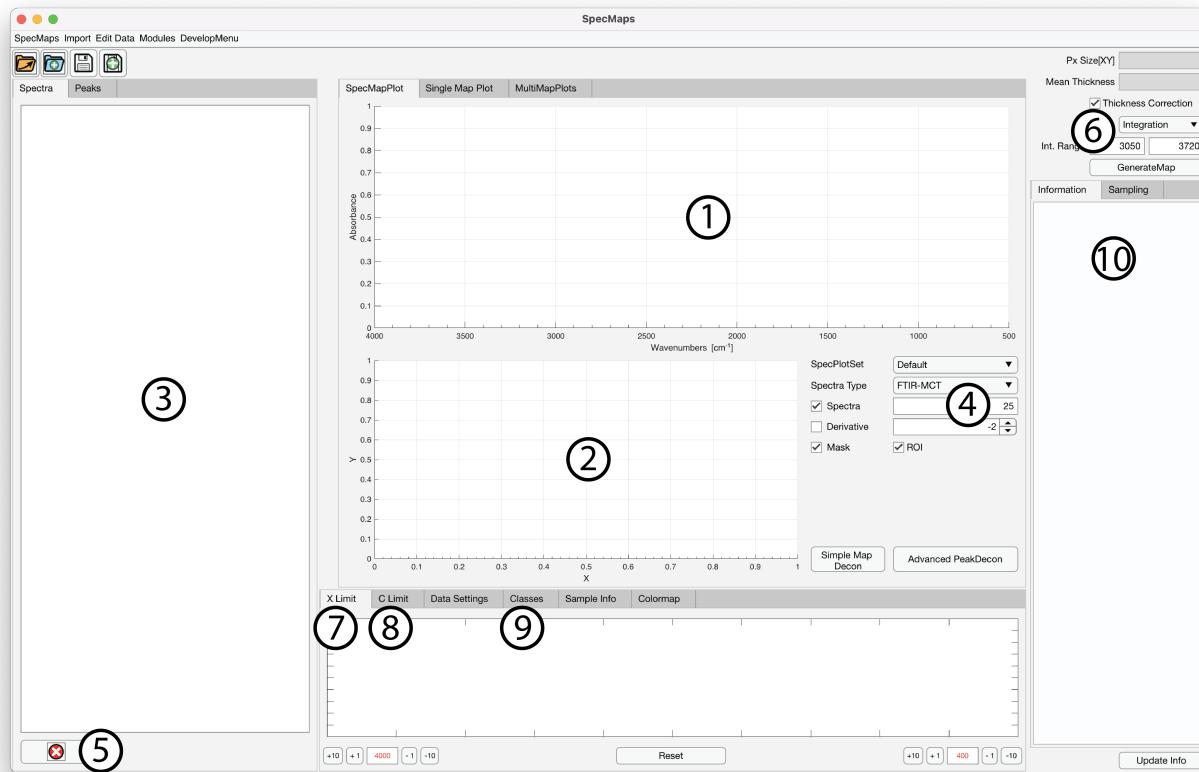


## 2 SpecMaps

SpecMaps is a graphical user interface (GUI) module for organising, manipulating, extracting, and comparing spatially resolved spectroscopic datasets. SpecMaps ist speziell für daten optimiert, deren spectra positionen in einem grid mit gleichen abständen gemessen wurden. For more details see input data.

### 2.1 SpecMaps GUI

The following section is dedicated to the most important components of SpecMaps and is intended to provide a basic introduction to the various key functionalities and tools for data analysis. 1-10 are shown in the figure SpecMaps GUI. 11-22 are self-explanatory.

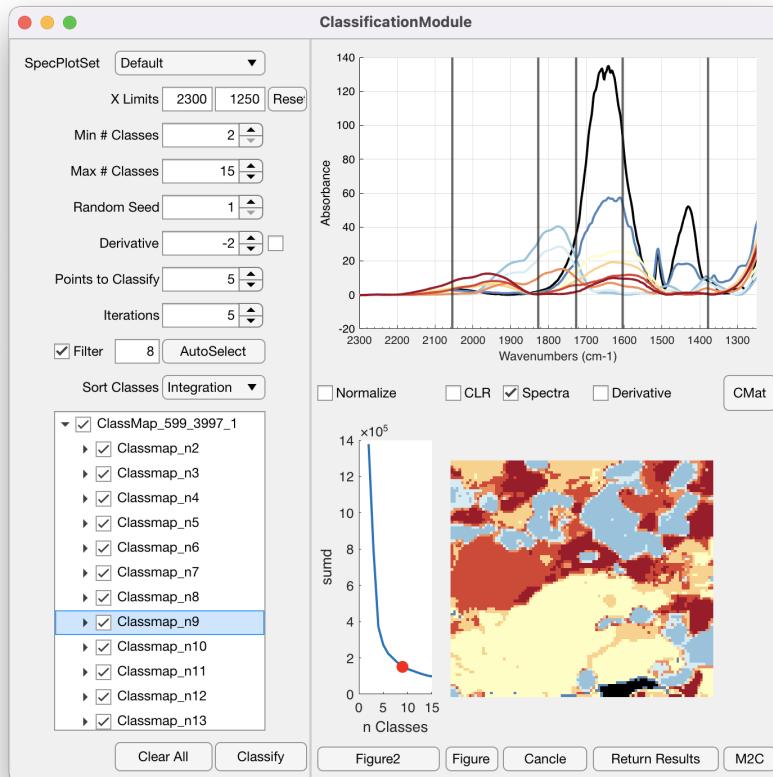


- |                               |                       |                               |
|-------------------------------|-----------------------|-------------------------------|
| 1. Spectra Plot               | 9. Classification     | 16.  Invert Mask              |
| 2. Map Plot                   | 10. Info window       | 17.  Reset Mask               |
| 3. DataTree                   | 11.  Automatic Import | 18.  ROI Circle               |
| 4. Number of preview spectra  | 12.  Save Project     | 19.  ROI Polygon              |
| 5. Delete selected            | 13.  Save Project As  | 20.  Single Pixel ROI Profile |
| 6. Map Generation Settings    | 14.  Create New Mask  | 21.  ROI Median Profile       |
| 7. XLimit - displayed x range | 15.  Add to Mask      | 22.  Delete ROI               |
| 8. CLimit - Colorbar limit    |                       |                               |

## 2.2 SpecMaps submodules

### 2.2.1 ClassificationModule

The classification module uses the unsupervised k-means algorithm, which performs the classification from the user input in "Min # Classes" to "Max # Classes". All results, as well as the initial raw data, can be sent to SpecMaps and the database module SpecDB, including all additional sample information.



### 2.2.2 ImportMatchModule

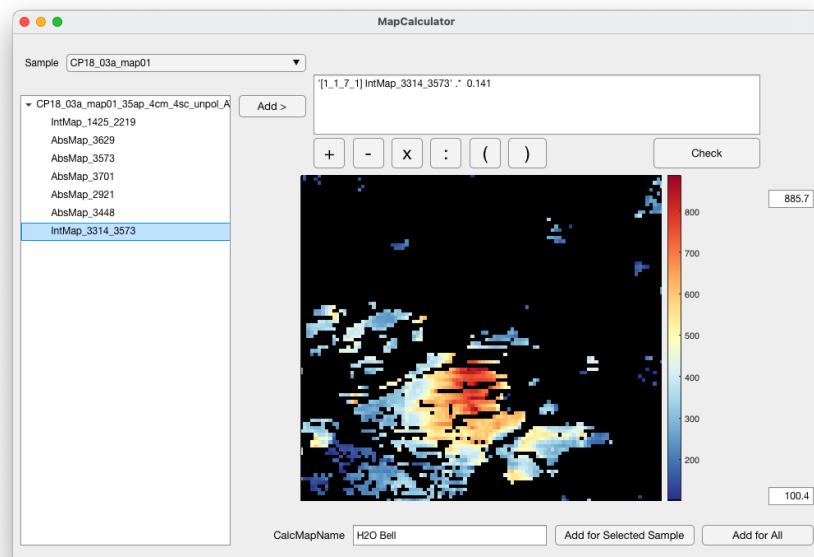
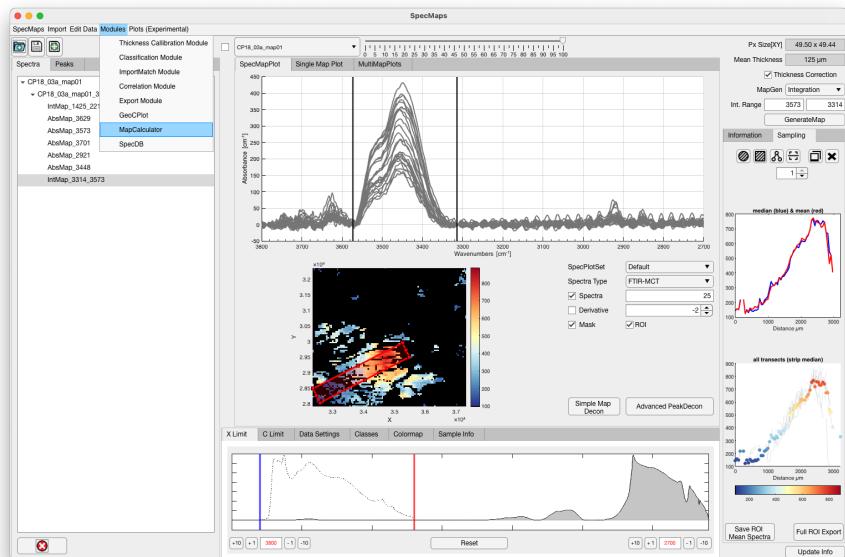
The ImportMatchModule allows the user to load external numerical data sets in the form of a matrix and reference them to the spectra map with a simple 3-point reference process. The input can either be a txt or csv file or a XmapTools project file. If the selected file is valid, the user must set 3 points on the reference map (spectra map) and then set the same 3 points in the map to be imported.

### 2.2.3 CorrelationModule

The correlation module allows a pixel correlation of all checked maps of a sample. A single classmap needs to be checked to colour the individual points with regards to the class they belong to. This module was in the first place created to be able to link spectra data to quantitative chemical data (e.g., EMPA, LA-ICP-MS). However, it turned out that it is also particularly useful to investigate relationships between various parts of a spectra and can be used to correlate different integrated absorbance, absolute absorbance, ratio maps, deconvolution parameter result maps or any other numerical data of the same area.

## 2.2.4 MapCalculatorModule

The MapCalculatorModule allows mathematical operations to be performed on the generated maps in a SpecMaps project, that share the same reference frame. It is also possible to apply MATLAB functions to generate new maps with custom calculations. The resulting new map can then be added to the SpecMaps project. If all samples and subsamples in a SpecMaps project share the same maps in the same order (e.g., generated by default maps for all samples), it is possible to perform the operation on all samples and subsamples.



## 2.2.5 ThicknessCalibrationModule

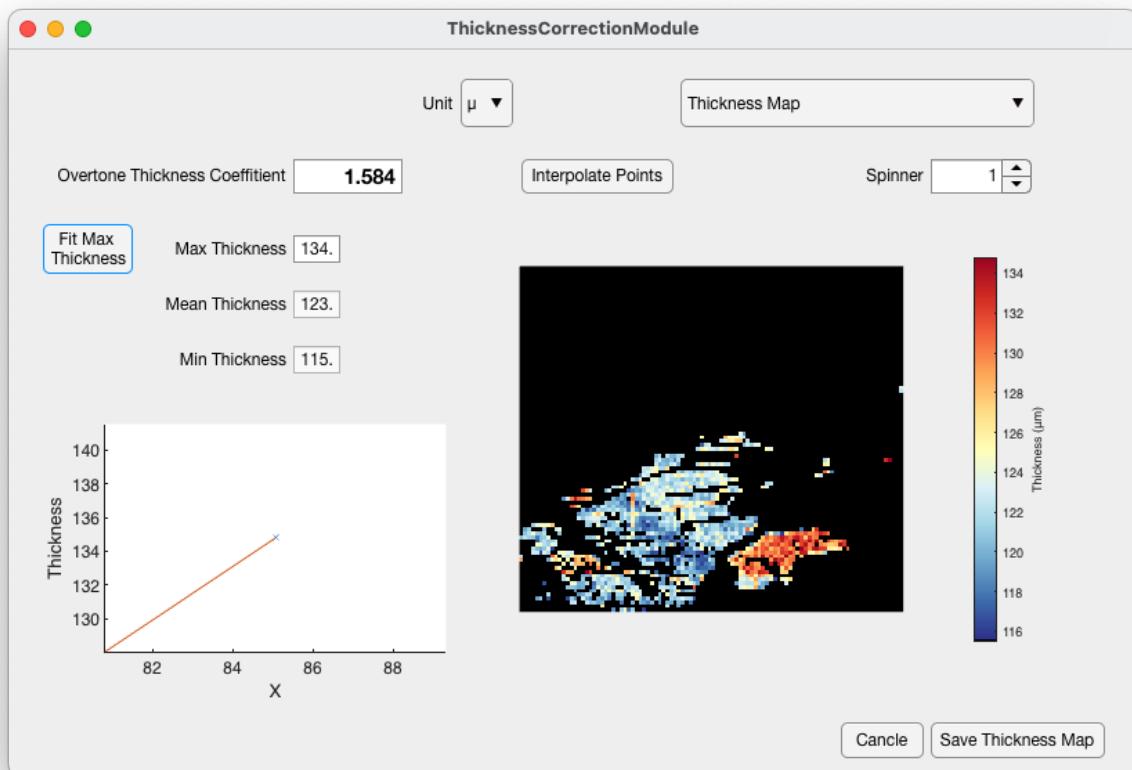
Disclaimer: There are known errors and limitations in the ThicknessCalibrationModule. Since it is one of the earlier modules it will be updated in the future to allow an individual calibration for each class or only a single class and then interpolate in between these classes to also cover the masked areas. Until then it is recommended to perform thickness calibrations with the MapCalculator (SpecMaps-Modules-MapCalculator).

This section describes the thickness correction for FTIR absorption spectra of single crystals to 1 cm thickness.

The ThicknessCalibrationModule allows the user to correct for changes in sample thickness and within the mineral of interest. To obtain a correct thickness correction, it is important that both the region of the spectrum used for the correction and the region to be corrected do not overlap with other significant peaks (exceptions: deconvoluted spectra, quant and norm maps). However, it should be noted that thickness correction should only be performed when spectra and sample are well known and even then, a critical evaluation of the results must be performed. Especially in the case of mixed spectra, micro-inclusions and interference effects, thickness correction can lead to errors that are difficult to trace, and a constant thickness value may be more appropriate.

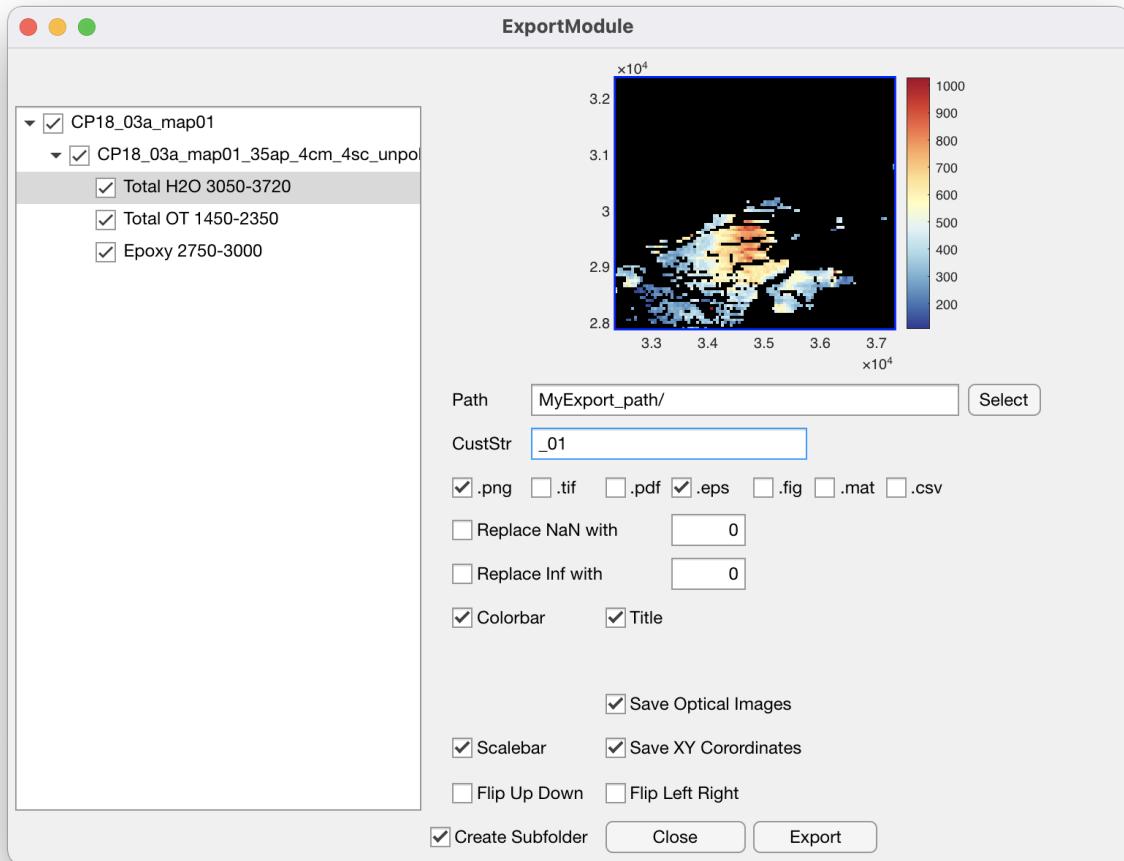
First a map for the calibration needs to be generated and selected which represents the thickness change in the map (e.g., the integrated Absorption of a Si-O overtone). This can be done in the map generation section (6) by typing in the extent of the desired region, ensuring the MapGen is set to integration and clicking Generate Map.

Selecting in the Edit Data-Thickness Calibration-Open Thickness Calibration opens the ThicknessCalibrationModule.



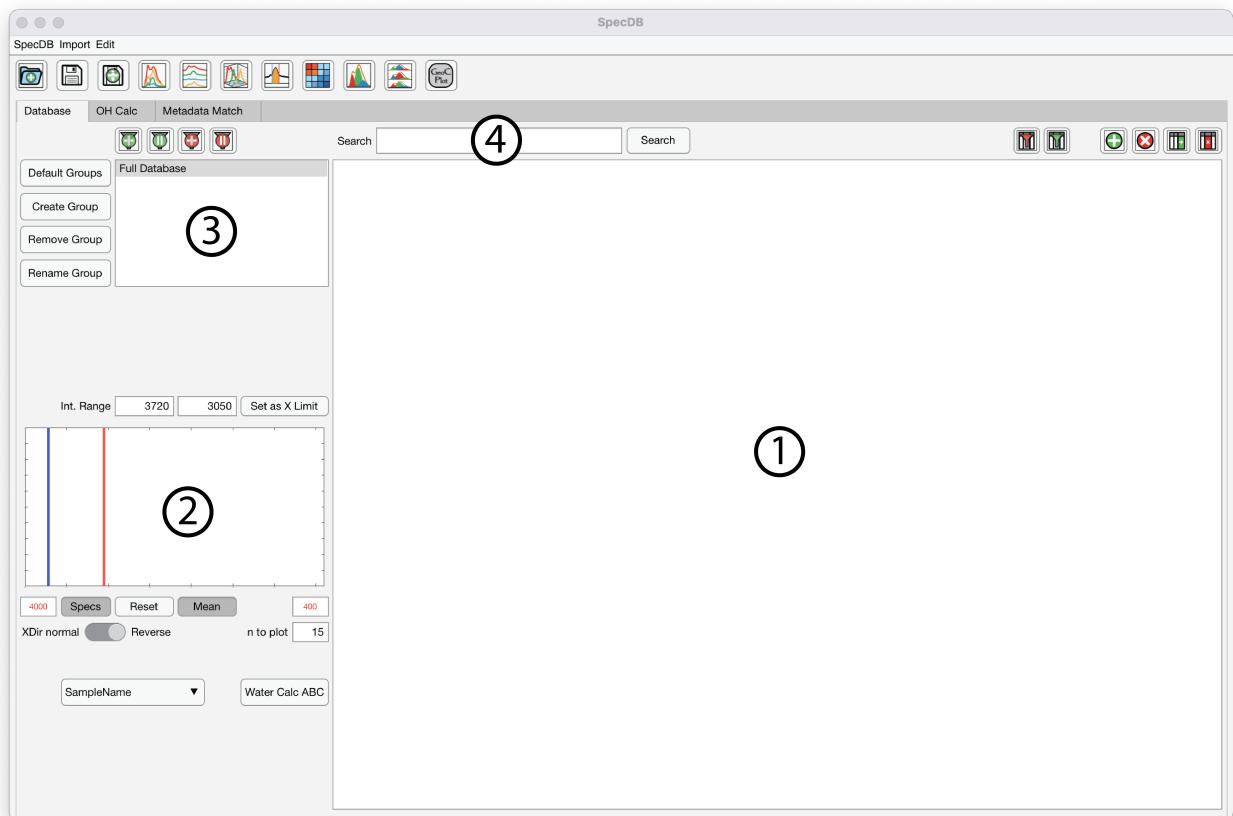
## 2.2.6 ExportModule

This module allows the user to export the generated maps for all samples in the most used formats (e.g., .png , .pdf, .eps, .txt, .fig, ...) for different purposes. The coordinates of all maps can be exported, to allow an easy import the maps into other software with the correct scale. For the image and vector graphics it is possible to add a colorbar and title and save the selected results in a folder structure equal to the data tree structure.



### 3 SpecDB

SpecDB is designed to organise, manipulate, evaluate, and store spectral data and results. Spectra can be imported either by direct import or as extracted spectra from SpecMaps projects. Metadata can be added to the database either by manual entry or by automatic matching to the sample name. Groups of spectra can be created and used to filter samples within the database. SpecDB can also extract spectra from different sources with varying number of data points and convert them to a uniform data format.



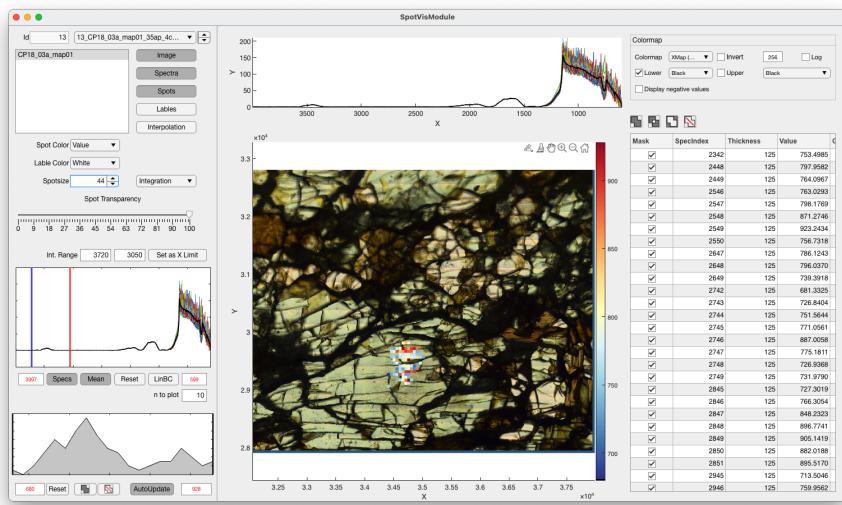
#### 3.1 SpecDB GUI

- |                         |                        |                          |
|-------------------------|------------------------|--------------------------|
| 1. Spectra Plot         | 9.  Simple Plot        | 17.  Delete ROI          |
| 2. Data Table           | 10.  Stacked Plot      | 18.  Filter Group &      |
| 3. Spectra Plot         | 11.  3D Plot           | 19.  Filter Group or     |
| 4. Groups Filter        | 12.  Extraction Module | 20.  Filter Group not &  |
| 5. Search String Filter | 13.  SpotVisModule     | 21.  Filter Group not or |
| 6.  Automatic Import    | 14.  SimpleDeconModule | 22.  Set Column Filter   |
| 7.  Save Project        | 15.  SpecVisModule     | 23.  Reset Column Filter |
| 8.  Save Project As     | 16.  GeoCPlot          |                          |

## 3.2 SpecDB submodules

### 3.2.1 SpotVisModule

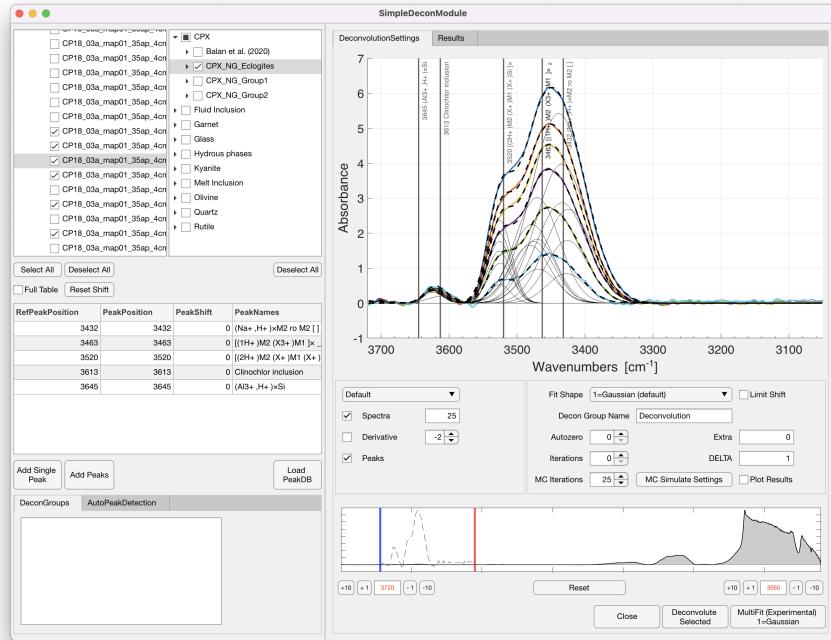
The SpotVisModule allows the visualization filtering and classification of xy referenced spectra. It is intended to be used for data that was not measured in a perfect grid. An individual thickness correction can be applied to each spectrum and a overtone thickness calibration can be performed. Deactivating the checkbox in table will deactivate the spectra from the calculation of the mean spectrum. Profiles, spots or spotmaps form individual groups can be edited and plotted and spot position and intensity of different extraction methods (e.g., integration, norm, etc.) can be visualized. The red and blue line in the left spectrum window mark the extraction range which is used to generate the value based on the selected method. The histogram shows the distribution of the values. Adjusting the black lines changes the colorbar scale and allows masking of the spectra outside of the selected range. This allows to filter inclusions and helps to generate the best possible mean spectra or profiles using the masking buttons. Additional features such as "plot profile" and export the value as well as x and y coordinates of each point can be accessed in the context menu by right clicking the table.



### 3.2.2 SimpleDeconModule

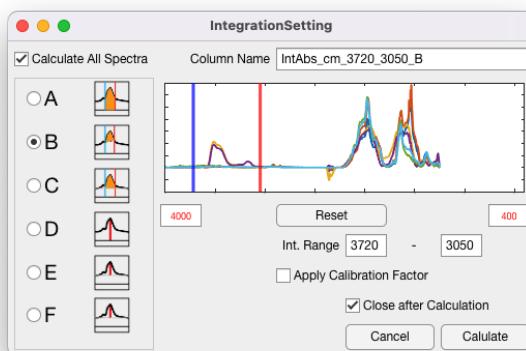
For all peak deconvolution calculations, SpecXY uses a modified version of the peakfit function (O'Haver 2022), which uses a simplex inversion to find the best-fit solution for given input parameters (e.g., number of peaks, peak shape, initial guesses; detailed information can be found in O'Haver, 2015; 2022). We modified the peakfit function to calculate the distance between the initial peak position estimates and the resulting fit position after each iteration of fit minimisation, and to return a high error if the value for any peak exceeds the given maximum shift. This allows the shift of each input peak to be individually limited to a given maximum and minimum value. When deconvoluting a spatially resolved dataset, peak shifts within the dataset can be visualised, showing the spatial distribution of peak area, peak position, and shift variations. The SimpleDeconModule provides a solution for performing Monte Carlo assisted peak deconvolution, calculating, and visualising the variation, and assessing the reproducibility of the peak deconvolution. The user can select initial peak positions either from a peak database or manually add user-defined peaks. The Monte Carlo assisted peak deconvolution performs the deconvolution for the specified number of iterations and saves the deconvolution result with the best R<sub>2</sub> values. These results are then plotted in histograms for each peak and parameter (position, half-width, area, height) and as iteration versus peak. For a given dataset Monte Carlo peak deconvolution allows estimation of the variability in the deconvolution results for a given initial guess and thus quantification of the possible impact on the calculations performed with the peak deconvolution results. This approach also helps to identify hidden

peaks and shifts for multiple spectra, which can then be used as initial guesses to perform an automatic deconvolution of each spectrum in a spatially resolved spectroscopic dataset. Once a satisfactory initial peak estimate is found, selected spectra from the database or spatially resolved datasets can be efficiently processed using the solution for the half-width and peak position solutions from the Monte Carlo assisted peak deconvolution as the initial starting estimate.



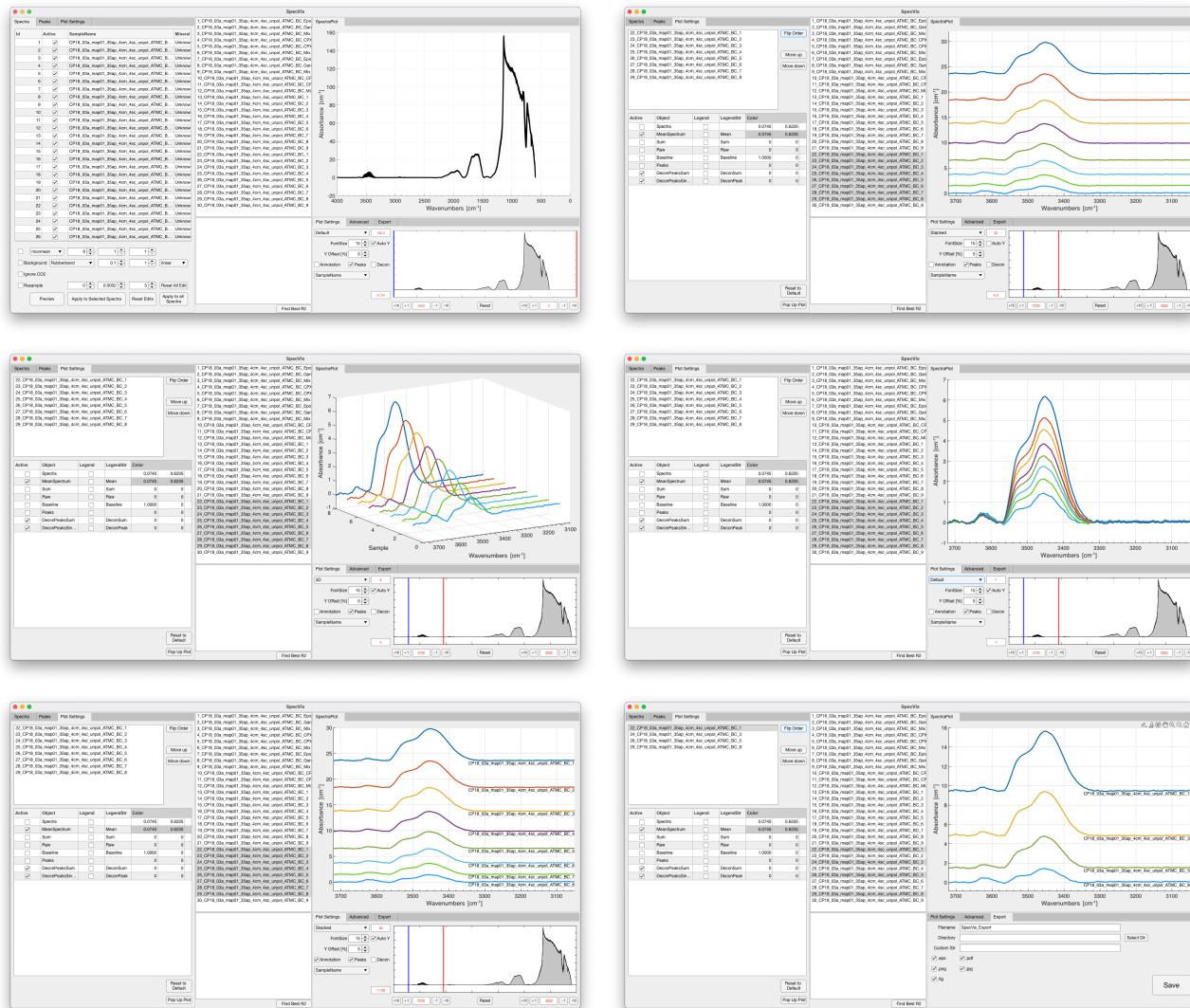
### 3.2.3 ExtractionModule

This module allows to extract numerical data from spectra. The data is then added as a new column to the SpecDB table. There are different options to depending on the chosen method. A-C are Integrating the spectrum between the red and blue line. D and E are extracting the absolute value at the red line. C and E performing a linear baseline correction before the extraction using the minimum value in the current view. B is performing a linear correction using the values at the edge of the current viewpoints of the current view. It is also possible applying a calibration factor to the extracted values. F is normalizing the visible range of the spectra from 0 to 1 and extracting the value of the chosen x position. D, E and F allow batch extraction of multiple comma, space, or tab separated values.



### 3.2.4 SpecVisModule

For visualisation of spectra and deconvolution results SpecDB includes the SpecVisModule. Spectra from the database can be plotted as normal, stacked or as 3D plots. The SpecVisModule allows quick modification of the generated plots to produce publication-ready figures, e.g. strings from the database can be added as annotations. Colours and appearance of the spectra, axis limits and label font size can be easily adjusted, and peak positions and names from a peak database can be added. All resulting plots can be exported as .eps, .pdf, .jpg or .png files.



## 4 GeoCPlot

GeoCPlot allows large datasets containing numeric values and strings to be examined in a tabular format and publication-quality correlation matrices to be produced. The dataset can either be loaded in from csv or excel files, or can be passed on from the database module SpecDB, which allows user to examine extracted numerical values from large spectral datasets. The dataset can be classified by unique strings or numerical entries of a column or using a semi-automatic k-means clustering method with selected columns containing numerical values. Selected columns containing numerical values, are then plotted in correlation matrices with a colour scale representing the class resulting from the classification. In addition, the mean, median values and all data points of all classes can be stored as plot groups. The appearance of all items in these groups can be modified and selected items will appear in the final plot. All generated plots can be exported as vector graphics.

## 5 Quick Start Guides

### 5.1 SpecDB

### 5.2 SpecMaps

#### Import

##### Autoimport

To automatically import Maps into SpecMaps select (11)  Automatic Import and select the folder containing the data. The folder name is used as Sample and the file name containing the spectra is used as subsample. Multiple maps stored in sub folders containing the following files can be loaded automatically:

- .mat files Opus (Matlab 4) export containing a matrix AB with
  - AB(:,1)=x\_signal
  - AB(:,2:end) =y\_signals
- \_mapinfo.txt annotation table from OPUS (must contain columns x, y, Group, Fileindex, Type)
- \_image.bmp (optional)
- \_image\_info.bmp (optional) image
- \_image2.bmp (optional)
- \_image2\_info.bmp (optional) image
- \_thickness.txt (optional) -> first line [min max mean] second line values (tab separated)

The automatic import function imports all available data with the described structure in the path specified by the user.

#### Generating the first Maps

To generate a 2-dimensional map from a multi-dimensional spatially resolved spectral dataset, each must be reduced to a single data value. After the data is loaded into SpecMaps the user can generate maps and investigate the sample. To generate a map the user must select a dataset in the DataTree(3) and press the GenerateMap button (6). The default settings for map generation are applied to generate an integration map and the result is added to the DataTree(3).

##### *Integration*

Set the integration area either by typing (6) or by moving the two black lines in the SpectraPlot (1).

##### *Absolute*

Set the x value of interest typing (6) or by moving the red line in the SpectraPlot (1) or type in the values in the boxes.

##### *Ratio*

Set the ratio values either by typing (9) or by moving the two lines in the SpectraPlot (1). Absolute

##### *Entropy*

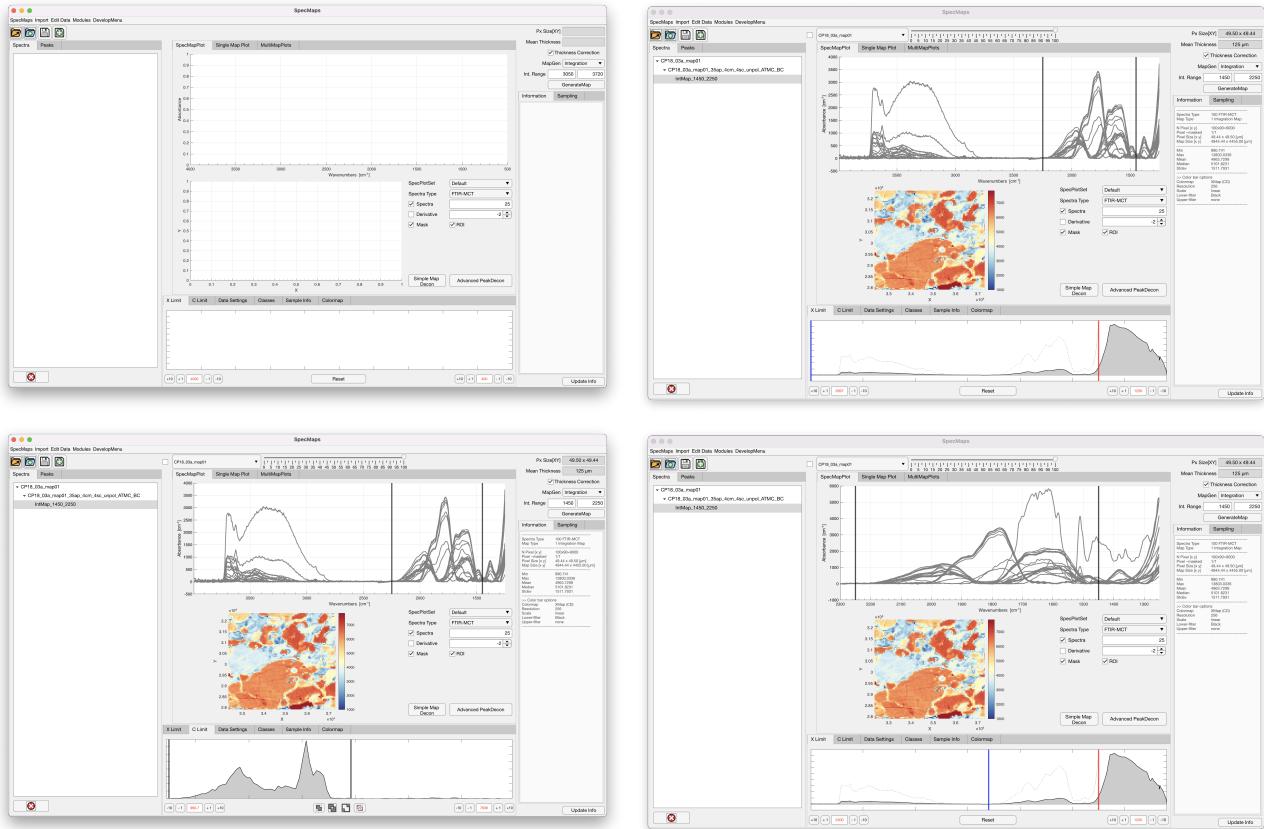
Signal noise 0 to 100 the higher the number the lower the signal noise

##### *Generate Map for all Samples*

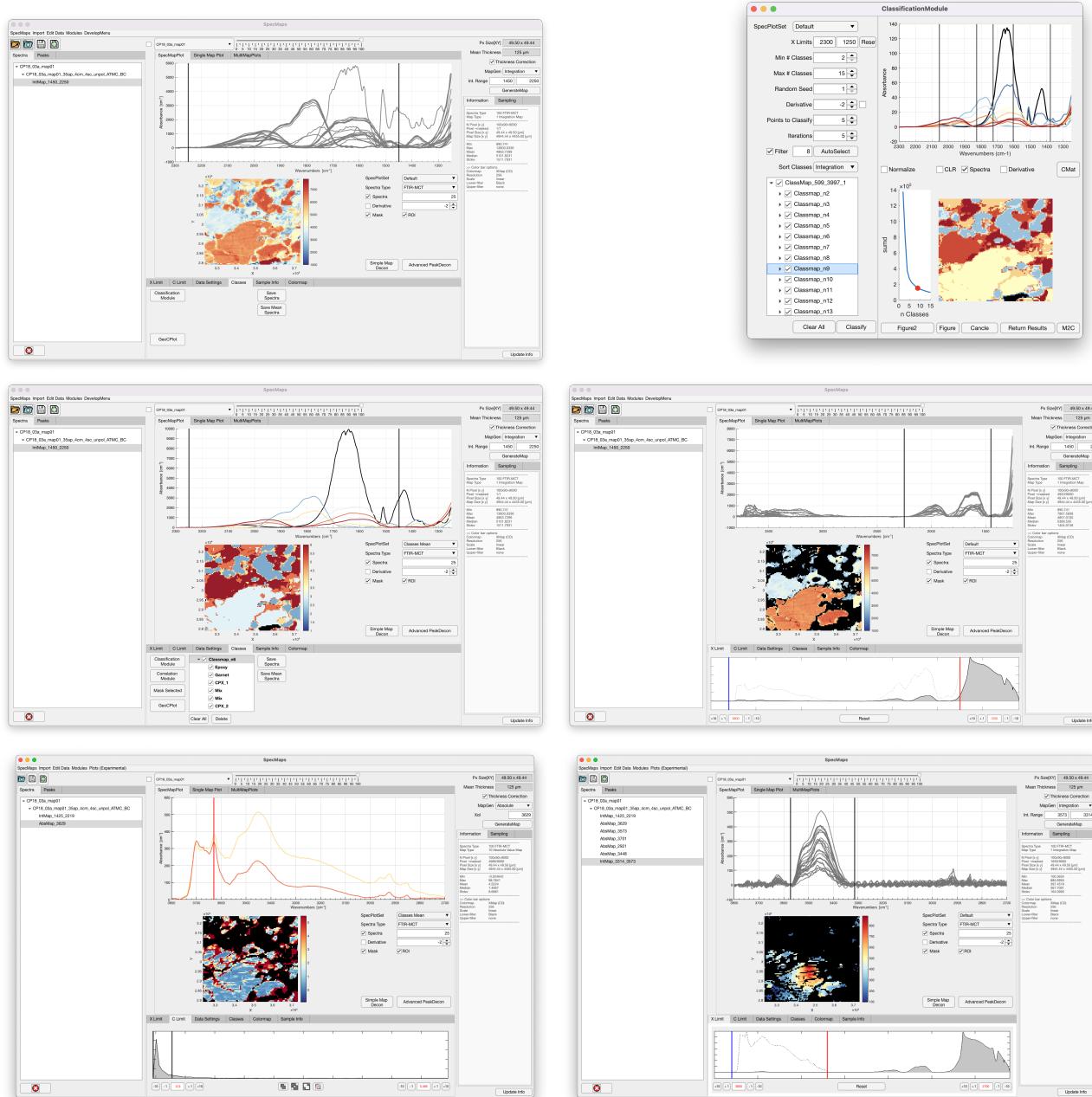
Create the map you want to generate for all samples for any sample. In the menu Edit Data-> Generate Map -> Generate map for all samples. SpecMaps will ask you for a name of the new map and generate the map with the settings chosen in (6) for all samples and subsamples.

##### *Preset Maps*

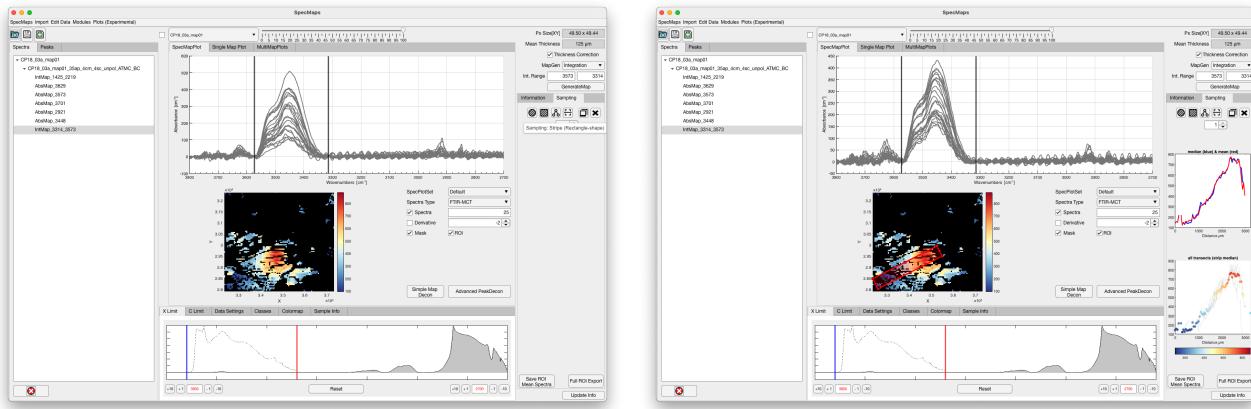
Groups of preset maps can be defined in the "MapPresetTable.csv" file which can be found in the installation directory.



- Load Data
- Generate Map
- Use histogram in CLimit tab to filter map
- Use XLimit tab to zoom in the overtone region



- Classes tab - Classification Module
- Chose bands to classify
- Classify
- Check classification results
- Return Results
- Mask Classes by checking - Mask Selected
- Use XLimit tab to zoom in the H<sub>2</sub>O region
- Generate Absolut maps to filter unwanted signals (e.g., sheet silicates, fluid inclusions, etc.)



- Generate H<sub>2</sub>O integration map
- Sampling tab (right panel)
- Stripe Sampling
- Extract profile
- Fill ROI Export

## 6 SpecXY API

The following chapter is for users running SpecXY inside MATLAB and gives an overview over possible public properties, functions and how to access and use them.

## 7 SpecXY Data Structure

(this section will be edited and a structure diagram added )

### 7.1 SpecXY Public

#### SpecXY

- **Description:** Start SpecXY.
- **Usage:** SpecXY\_public = SpecXY;

#### SpecMaps Data

- **Description:** Access SpecMaps Projects Data.
- **Usage:** SpecXY\_public.SpecMaps\_public(n).Data;

#### SpecDB Data

- **Description:** Access SpecDB Data.
- **Usage:** SpecXY\_public.SpecDB\_public.Data;

### 7.2 SpecDB Public Functions

#### insert\_data

- **Description:** Insert data into SpecDB.
- **Inputs:**
  - SpecDB: SpecDB instance.
  - data: Structure with x\_signal, y\_signal, and Meta information.
  - update\_set: Boolean flag to update UI and table.
- **Usage:** insert\_data(SpecDB, data, update\_set);

#### get\_specs

- **Description:** Retrieve spectra from SpecDB.
- **Inputs:**
  - SpecDB: SpecDB instance.
  - selections: Unique Ids from data table.
  - mask\_specs (optional): Mask spectra.
- **Outputs:**
  - x, y, SampleName, Thickness: Cell arrays with {n\_selections}.
  - Id, eID: Unique Ids and positions in data table.
- **Usage:** [x, y, SampleName, Thickness, Id, eID] = get\_specs(SpecDB, selections, mask\_specs);

### **CalcIntAbs\_cm**

- **Description:** Calculate and extract values from spectra.
- **Inputs:**
  - SpecDB: SpecDB instance.
  - settings\_return: Settings for calculation.
- **Outputs:** answer: Results of the calculation.
- **Usage:** answer = CalcIntAbs\_cm(SpecDB, settings\_return);

### **MatchWithDB**

- **Description:** Experimental function for matching spectra.
- **Inputs:**
  - SpecDB: SpecDB instance.
  - x\_signal, Y\_signal: Signals to match.
  - n\_match: Number of matches to find.
  - mmv\_input, mmv\_DB: Moving mean window (experimental).
  - baseline\_cor: Baseline correction flag.
- **Outputs:** results\_id: Selected Ids based on matching.
- **Usage:** MatchWithDB(SpecDB, x\_signal, Y\_signal, n\_match, mmv\_input, mmv\_DB, baseline\_cor);

### **update\_public**

- **Description:** Update SpecDB UI or table.
- **Inputs:**
  - SpecDB: SpecDB instance.
  - what2update: 0 (update UI), 1 (update DB table), 2 (update DB list).
- **Usage:** update\_public(SpecDB, what2update);

## **7.3 SpecMaps Public Functions**

### **ThicknessMapReturn**

- **Description:** Insert Thickness\_map into SpecMaps.
- **Inputs:**
  - SpecMaps: SpecMaps app instance.
  - Thickness\_map: Thickness map data.
  - sampleinfo: Information about the sample.
- **Usage:** ThicknessMapReturn(SpecMaps, Thickness\_map, sampleinfo);

### **ClassificationReturn**

- **Description:** Insert Classification into SpecMaps.
- **Inputs:**
  - SpecMaps: SpecMaps app instance.
  - Classification: Classification data.
  - sampleinfo: Information about the sample.
- **Usage:** ClassificationReturn(SpecMaps, Classification, sampleinfo);

## **MapMatchReturn**

- **Description:** Insert referenced and interpolated numerical matrices (maps).
- **Inputs:**
  - SpecMaps: SpecMaps app instance.
  - matrix\_to\_match: Numerical matrices to match.
  - matrix\_to\_matchNames: Names of matrices.
  - sampleinfo: Information about the sample.
  - type: Type of data (e.g., Quantdata).
- **Usage:** MapMatchReturn(SpecMaps, matrix\_to\_match, matrix\_to\_matchNames, sampleinfo, type);

**Various other functions for plotting, updating, and map manipulation...**

## **7.4 Public SpecDB Properties**

- **Data:** Data structure including all data.
- **edit\_state:** Default false, set to true if there is a change in the project since the last save.

## **7.5 Public SpecMaps Properties**

- **SpecMap:** Data structure including all data.
- **ColorMaps, ColorMapValues, ColorScale:** Color map properties.