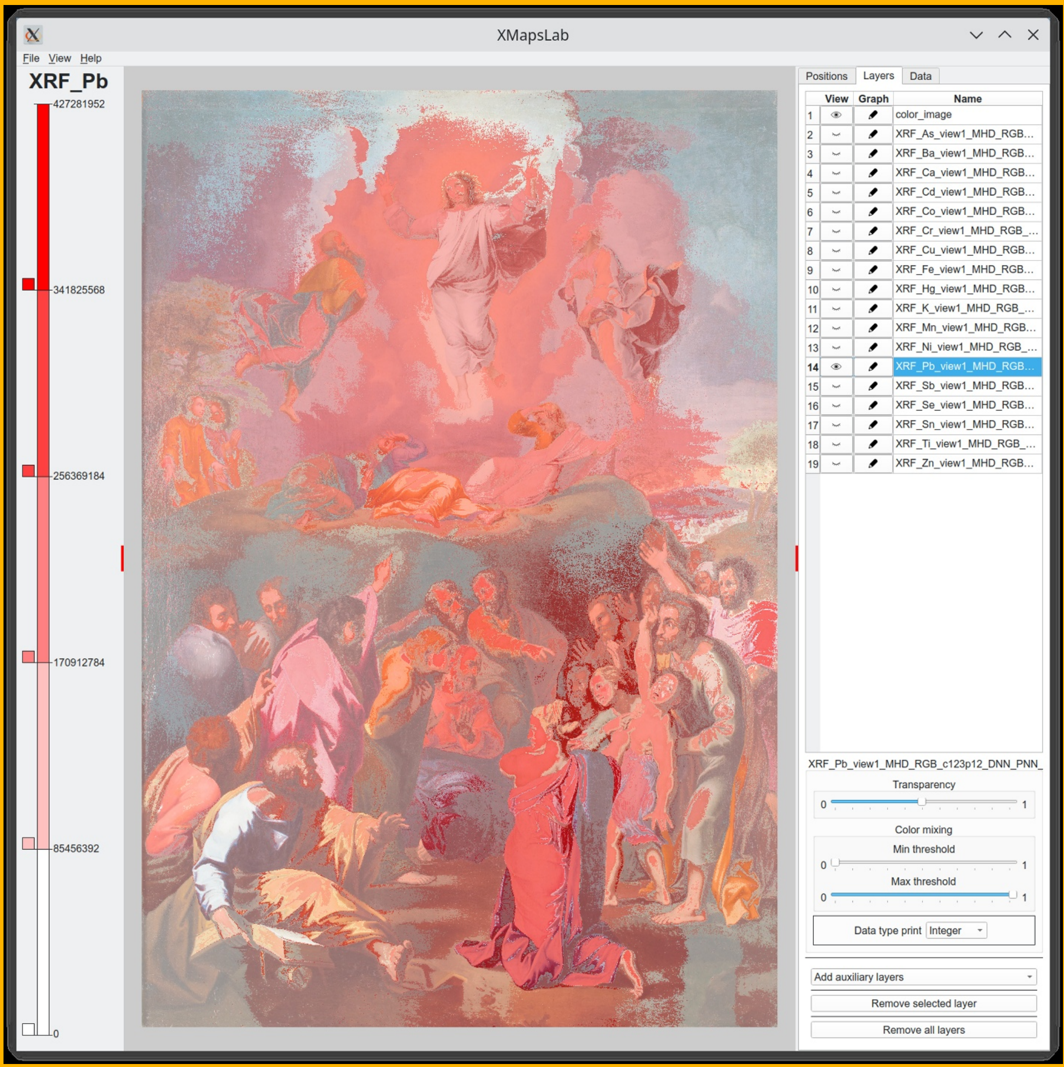


XMapsLab manual v.2

Domingo Martín Perandrés
dmartin@ugr.es
https://calipso.ugr.es/xmapslab.org

2025

Contents

List of Figures

1

Introduction

XMapsLab consists of a set of programs designed to facilitate work with XRF, XRD, and other types of spectral data. Its primary objective is to extract relevant information about elements and pigments from spectral data acquired through scanning devices, as well as to generate graphical representations illustrating the distribution of these elements and pigments.

Currently, we have developed the program **XMapsLab** which enables the generation of maps from validated data. This data comprises elements and/or pigments extracted from the raw outputs of scanning devices and subsequently validated by experts in material sciences, painting conservation, and restoration. Additionally, we have created **Positions** a program that assists in defining measurement positions.

1.1 The General Process Steps

The following steps outline the procedure for studying an artwork, such as a painting, using the XRF technique to produce maps:

1. Acquisition of a high-quality photograph
   The first step is to obtain a high-resolution photograph of the artwork. This requires a high-quality camera and lens, proper positioning and orientation, and uniform lighting. The photograph should capture sufficient detail and be processed to enhance the extracted information. It is recommended to save the images in RAW format and adjust the white balance and colors, ideally using a calibrated color reference. Once processed, the image should be saved in a lossless format with an alpha channel option, such as PNG.

   Regarding the appropriate size for the maps, there are no strict rules. **XMapsLab** supports real-time or interactive work with relatively large images, for example, 3000 $\times$ 4000 pixels.

2. Selection of measurement positions
   After acquiring the photograph, the measurement positions must be determined. There are two possible approaches:

   - Direct selection on the artwork: Measurement positions are marked non-invasively on the artwork itself and later input into the project.
   - Selection via a photograph: A high-quality photograph of the artwork is used to mark measurement positions digitally with a program such as **Positions**. The resulting annotated image can be printed or displayed on a mobile device, allowing experts to review and refine the selection of measurement points.

3. Data capture
   Once the measurement positions have been established, the scanning device is used to acquire the data.

4. Data processing and validation
   The scanning device generates data files that must be processed and validated by the user to extract information about the elements and/or pigments present in the artwork.

5. Analysis and mapping with **XMapsLab**
   The validated data, combined with the measurement positions and the high-quality photograph, serves as the basis for studying and analyzing the artwork using **XMapsLab**.

It is important to note that this process assumes manual scanning at a limited number of positions, which is common when using handheld devices. In cases where the scanning device features mechanized positioning, interpolation methods are generally unnecessary, and the measurement positions are determined directly within the software that controls the scanning system.

1.2 Using XMapsLab

To analyze an artwork using **XMapsLab** a dedicated project must be created. This project should encompass all relevant information required for studying the artwork, particularly for generating maps.

The project must include the following data:

- A high-quality color photograph of an appropriate size.
- The precise positions where measurements were taken.
- The corresponding spectral data obtained from those positions.

All this information is organized within a project structure consisting of a hierarchical folder system and multiple files. The specifics of this structure will be detailed in the following chapter.

2

Creating a project for XMapsLab

2.1 The project

To work with **XMapsLab** a project must be prepared, which consists of a structured set of folders and files. The project contains all relevant data related to the studied artwork, ensuring a systematic organization for analysis.

The structure of a project is illustrated in Figure 2.1. The main folder, which houses the programs, contains two subdirectories and two programs:

- program_data: Stores program-related data.
- projects: Stores individual projects, each represented as a separate folder (e.g., project_1). Typically, project folders are named after the artwork being analyzed.
- Positions: The program the helps to define the measurement positions using a photography of the artwork.
- XMapsLab: The program that produces and analyses the maps.

The projects folder must be shared between the **XMapsLab** and **Positions** programs. Ensure that you correctly link or share this folder within your operating system.

It is crucial to maintain the specified folder structure and naming conventions, as the programs rely on these predefined paths to locate data correctly. The folder hierarchy and required files can be created manually using standard operating system procedures and any text editor.

Below is a description of the contents of each folder within a project:

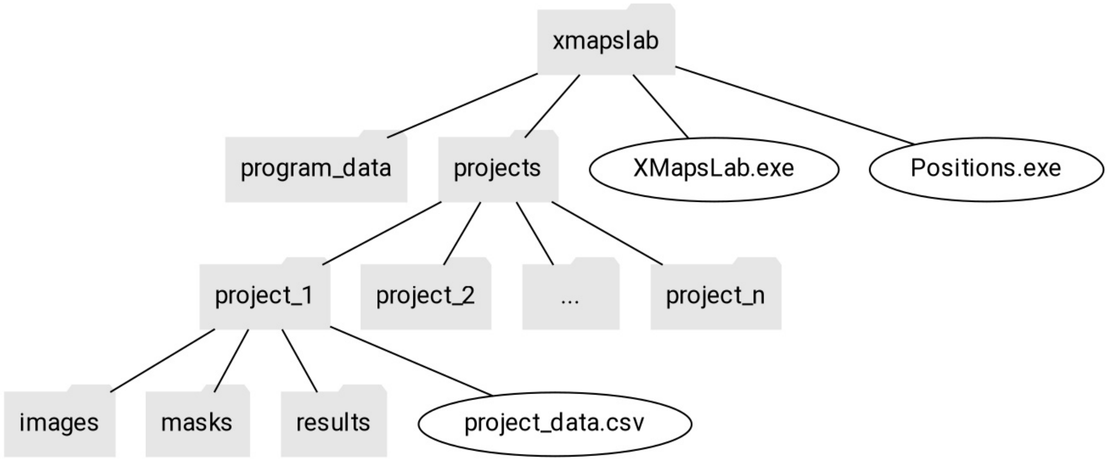


Figure 2.1: XMapsLab hierarchy

- images
  This folder contains the image file of the artwork. The image serves as a visual reference for the distribution of elements and/or pigments and is essential for generating maps using the MHD method (Martín-Ramos and Chiari [2019]). The recommended format is PNG, although JPG is also supported. The PNG format allows for the inclusion of a transparency channel, which can be used to exclude specific areas of the image from processingespecially useful for non-rectangular artworks. Multiple images can be stored in this folder; however, only one can be loaded into the program at a time.

> For optimal performance, image dimensions should be multiples of 4.

- masks (optional)
  This folder contains images of mask IDs and mask shapes, which may be used for additional processing. Currently under review.

- results
  This folder stores output files, including position images and generated maps.

- projec_name.csv
  The file that contains all the measured information. Multiple files can be stored in this folder; however, only one can be loaded into the program at a time.

Apart from the folder structure, the most important file to be able to run **XMapsLab** is the project_name.csv file. The information it should contain and how it can be created is described below.

2.2 The project data file

In addition to the folder structure, the most crucial file required to run **XMapsLab** is the project_name.csv file. The project_name component can be any name, but the .csv extension is mandatory. This file contains all the essential information required by **XMapsLab** to generate maps, including general details such as the project or artwork name, the positions where measurements were taken, and the values of various elements and/or pigments that have been measured and validated by experts.

Below is an example of a project_name.csv file:

```
XML;1.0.0
PROJECT_NAME;The Transfiguration
AUTHOR;DMP
DATE;18/05/24
DEVICE;XRF;DUETTO1
TUBE;XRF;CU
SPOT_SIZE;XRF;3
WIDTH_CM:62.3
HEIGHT_CM:93.2
WIDTH_PIXEL;2696
HEIGHT_PIXEL;3668
CS_ORIGIN;TOP_LEFT
POSITION;-;1;2;3;4;5;6;7
X;-;475;1273;1546;2365;2190;1294;1390
Y;-;2269;564;1369;2083;2512;2890;1844
XRF;Fe;0;62,344696;1314,839355;1685,118774;2614,584229;2015,744141;30,807642
XRF;Hg;926,384766;911,458984;3,695817;811,449341;7,174562;5,541962;776,76532
XRD;Calcita_Co;80,279877;259,103699;5,858643;291,345825;94,898193;97,665283;30,607697
```

A key aspect to note is that certain words, displayed in blue, are keywords and must be capitalized exactly as shown. These keywords define the meaning of the subsequent data. For example, after AUTHOR, the authors name should be provided. In some cases, such as in the previous example, the text can be freely defined (following specific formatting rules discussed later), whereas in others, predefined options must be selected.

Although this information can be entered manually, certain dataparticularly measurement positionsare more conveniently handled using a dedicated support program such as **Positions**.

Required fields in the project_name.csv file:

- XML
  This is the first required field. It identifies the file as an **XMapsLab** project file. The subsequent parameter specifies the version number, currently "1.0.0".
- PROJECT_NAME
  Stores the name of the project.
- AUTHOR
  Specifies the name of the projects author.

- DATE
  ndicates the date of file creation (or any relevant date chosen by the author). The format can be any valid date format, such as DD/MM/YYYY.
- DEVICE (optional)
  Specifies the device used for data acquisition. The second column must include the input type (XRF, XRD, or RAMAN), while the third column provides a descriptive text about the device.
- TUBE (optional)
  Describes the type of tube used in the device. The second column must specify the input type (XRF, XRD, or RAMAN), and the third column should contain a descriptive text indicating the tube type (e.g., copper, CU, or cobalt, CO).
- SPOT_SIZE (optional)
  Describes the spot size of the device: the diameter of the circle where the measurement is done. The second column must specify the input type (XRF, XRD, or RAMAN), and the third column should contain the diameter in millimeters.
- XRF_DATA_ADJUSTMENT (optional)
  If XRF data is included, this field indicates whether adjustments should be applied. Possible values are:
  - ATOMIC_NUMBER
  - ATOMIC_WEIGHT
  - NONE (no adjustment applied)

  If this field is absent, no adjustment is performed.
- WIDTH_CM
  Defines the width of the input image in centimeters.
- HEIGHT_CM
  Defines the width of the input image in centimeters.
- WIDTH_PIXEL
  Defines the width of the input image in pixels.
- HEIGHT_PIXEL
  Defines the height of the input image in pixels.
- CS_ORIGIN
  Specifies the origin of the coordinate system. The possible values are: TOP_LEFT or BOTTOM_LEFT.
- POSITION (optional)
  An auxiliary field indicating the numbering of measurement positions. The format consists of the POSITION identifier followed by a hyphen (-), then the position numbers (starting from 1), all separated by the defined separator character.

- X
  Contains the X coordinate for each measurement position. The format is the same as for POSITION. The coordinates must be normalized. This can be done manually by dividing the measurement coordinates by the corresponding dimension, i.e., X/WIDTH_CM and Y/HEIGHT_CM. Alternatively, the **Positions** program can be used, which automatically computes the normalized coordinates.

- Y
  Contains the Y coordinate for each measurement position. The same conditions are used as with X field.

- XRF or XRD or RAMAN
  Stores data from XRF, XRD, or Raman measurements. The format includes:

  1. The identifier (XRF, XRD, or RAMAN)
  2. The element or pigment name (e.g., "Fe", "K", "Hg", "Cinnabar", "Lapis Lazuli", etc.)
  3. The corresponding values, separated by the chosen separator character.

  Values can be real or integer numbers, but they are processed internally as real numbers. If no data is available for a specific position, a 0 should be recorded.

  The number of values should match the number of positions (X and Y coordinates). If fewer values are provided, missing entries will default to 0. If more values are included than the number of positions, the excess data will be ignored.

> The names of XRF elements must be written accurately to ensure correct atomic number and weight calculations.

> Since **XMapsLab** allows users to enable and disable selected positions dynamically, it is advisable to collect more data points than necessary during the scanning process. This approach minimizes the risk of missing data when conducting the analysis. Unnecessary positions can be easily disabled later rather than identifying missing measurements retrospectively.

The CSV (Comma-Separated Values) file is a plain text file that organizes tabular data using a specific character (typically a comma) as a delimiter. This format is widely used because:

- It can be opened and edited with a standard text editor.
- It is compatible with spreadsheet software such as Microsoft Excel and LibreOffice Calc.

- Spreadsheet programs can export data in CSV format, ensuring compatibility across platforms.

Given its simplicity and universal support, CSV is an ideal format for storing and managing project data in a structured manner.

2.2.1 Entering data using a spreadsheet

Using a spreadsheet is the most efficient, convenient, and error-resistant method for creating the project_name.csv file. The data should be structured as explained in the previous section. When saving the file, it is essential to select the semicolon (;) as the separator.

2.2.2 Entering data using a text editor

Alternatively, a text editor, such as Microsoft WordPad on Windows, can be used without requiring additional software installation. While text editors are user-friendly and widely available, they are primarily designed for text input rather than structured numerical data, making tabulation cumbersome.

Nevertheless, we will outline how to create a basic example while addressing key challenges that may arise.

```
XML;1.0.0
PROJECT_NAME;Test
AUTHOR;DMP
DATE;18/05/20
DEVICE;XRF;XRF pistol
TUBE;XRF;CU
SPOT_SIZE;XRF;8
XRF_DATA_ADJUSTMENT;ATOMIC_NUMBER
WIDTH_CM:50
HEIGHT_CM:50
WIDTH_PIXEL;1000
HEIGHT_PIXEL;1000
CS_ORIGIN;TOP_LEFT
POSITION;-;1;2;3
X;-;475;1273;1546
Y;-;2269;564;1369
XRF;Fe;0.0;12.34;20.45
XRD;Cinnabar;98.4;10.12;37.45
```

To begin, open a text editor and create a new file. Save it as project_name.csv.

The file must be saved as plain text without formatting. Any additional formatting can introduce unintended characters that may cause errors when reading the file.

Why plain text is essential

Text editors allow modifying the appearance of text. For instance, you might write "XML" and apply bold formatting, which visually changes how it appears:

```
XML
```

However, if saved with formatting, the actual file contents might include hidden control characters:

```
<bold>XML<\bold>
```

or even encoded representations such as:

```
?A‘XML?A>
```

These additional characters can render the file unreadable for **XMapsLab**. Therefore, the project_name.csv file must be saved strictly as plain text.

Manually Entering data

Now let's enter each line. Place the cursor on the first line, at the far left (which is the default position), and type XML in all capital letters. Then, add 1 separator, which is a semicolon (;), and after that, you can add the text "1.0.0". Since we will be writing more lines, press the return or new line key.

It is important to understand the writing rules:

- A separator cannot be placed at the beginning of an identifier.
- Only one separator (;) should be used between two values.
- No separator should be placed after the last value in a row.
- Only semicolons (;) can be used as separatorsdo not use commas, periods, spaces, tabs, or other characters.
- Each line must end with a newline (Enter key).

As mentioned earlier, it is crucial to be careful when writing the file and follow all the provided rules to avoid creating files that may look correct but cannot be read by the program.

While the information in the file remains the same, the way it is displayed varies depending on the software used. In a spreadsheet (Figure 2.2(a)), data is presented in a table format, where rows and columns are clearly distinguished. In a text editor (Figure 2.2(b)), data appears as plain text with semicolons separating values. Rows with fewer values may appear with additional semicolons to maintain alignment. For example, if the maximum number of columns in a dataset is 5, but some rows contain only 2 values, the missing columns must be indicated with 3 additional semicolons.

> Manually calculating and adding the required number of semicolons can be error-prone, especially when dealing with many columns. It is strongly recommended to use a spreadsheet for creating and editing the file.

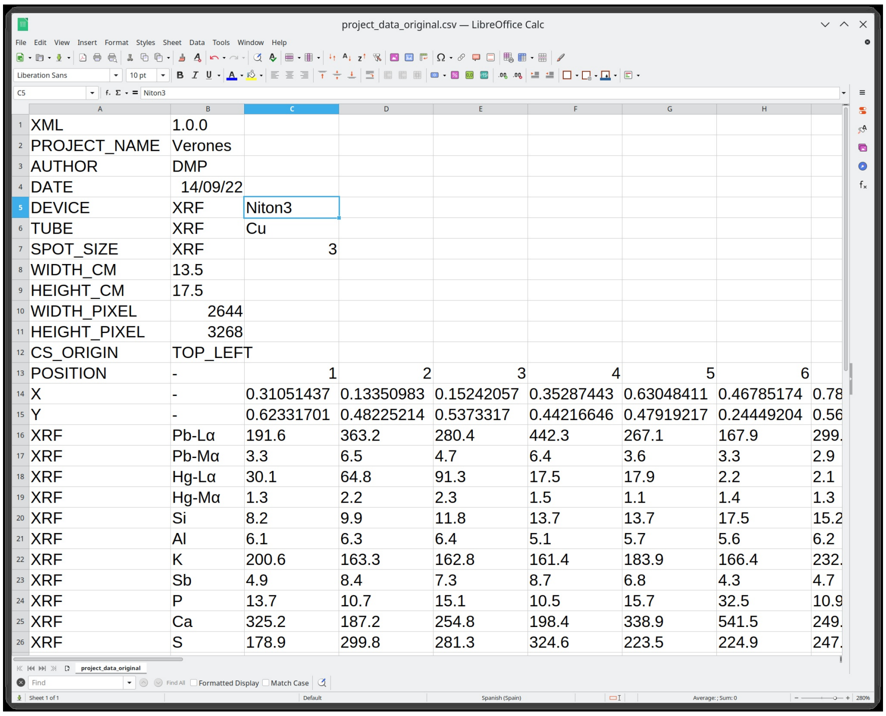
This example also illustrates how numeric data can be stored as integers or real numbers, with the option to use either a period (.) or a comma (,) as the decimal separator.

With the explanations provided, the process for entering position data and element/pigment values should now be clear:

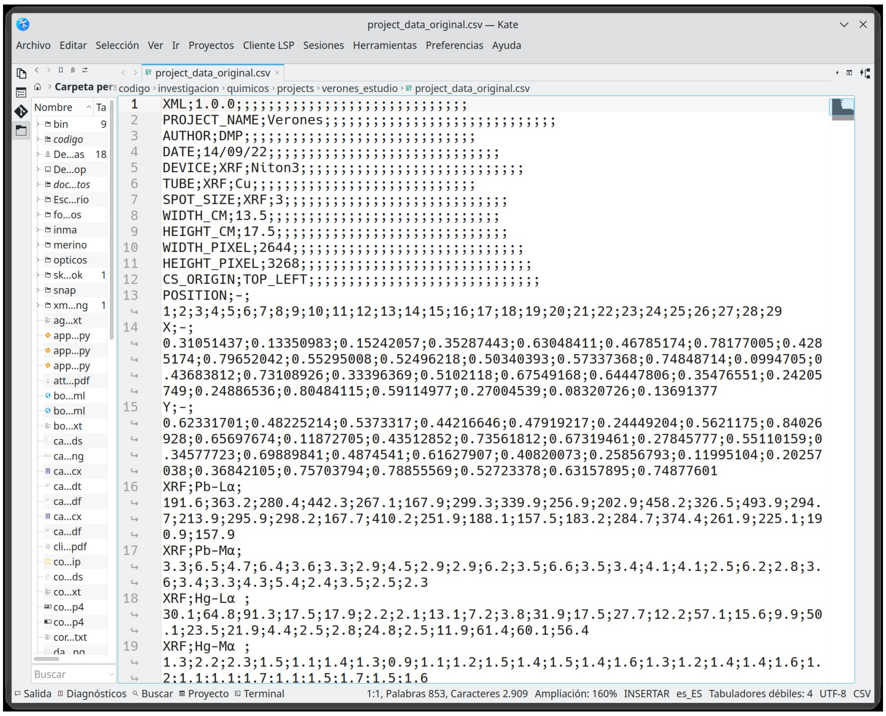
- For X and Y coordinates, a hyphen (-) must be placed after the identifier to ensure proper alignment with the corresponding elements.
- When entering measured data, begin with the identifier, followed by the element or compound name, and then input the respective values.

Although positions can be entered manually, a more efficient method is to use **Positions** which automates this process. For detailed instructions on using this program, refer to **Positions** in chapter 3.

Element and pigment values are typically inserted through a simple copy-and-paste operation (with transposition) from the spreadsheet generated by the scanning device into the spreadsheet prepared for **XMapsLab**.



(a) Spreadsheet



(b) Text editor

Figure 2.2: Aspect of the project_name.csv file in a spreadsheet and a text editor.

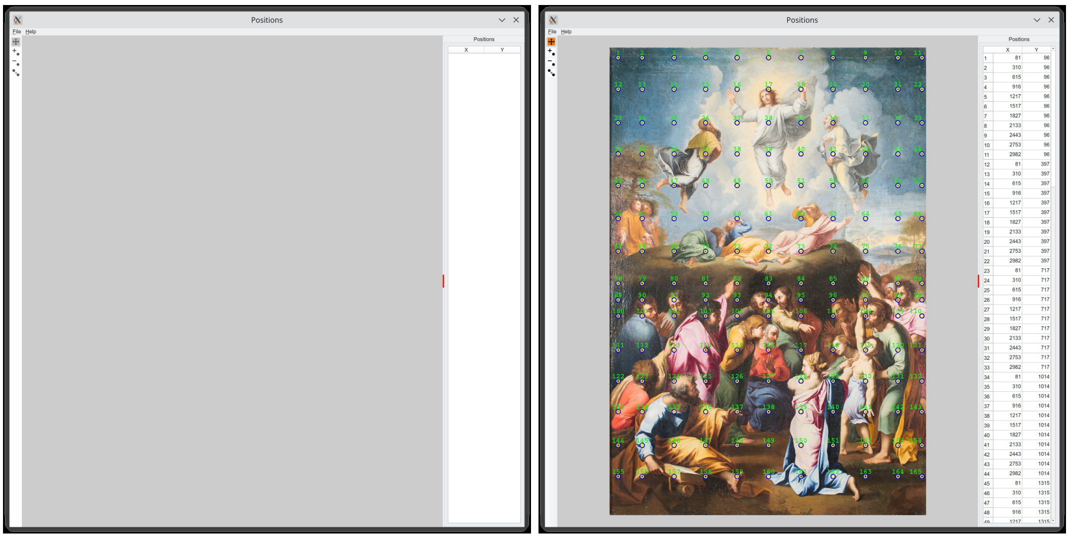
3

Positions

One of the most critical tasks is entering the coordinates of measurement positions in the project_name.csv file. While this can be done manually, it is a tedious and error-prone process, especially when dealing with a large number of positions. To streamline this task, the **Positions** program (Figure 3.1) provides an interactive graphical interface for defining measurement positions.

The program allows users to:

- Define measurement positions graphically and interactively.



(a) Initial state (b) Running state

Figure 3.1: Two captures of the **Positions** program in the initial state and when the positions are introduced.

- Save position data directly to the project_name.csv file.
- Generate an image with marked positions, which can be printed and used as a reference for positioning the scanning device.

As shown in Figure 3.1, the program interface consists of: A toolbar (left panel), an image display area (center panel) and a position information panel (right panel). The initial state upon launching the program is depicted in Figure 3.1(a).

The first steps involve selecting: Thee project to work with, the image that will be used to define measurement positions and the project data.

To select a project:

1. Navigate to the File menu and select Load project.
2. Choose the desired project from the displayed list.

If the project contains multiple images, the program will prompt the user to select one. Similarly, if the project includes multiple data files, the user will be asked to choose the relevant file.

Once the project data is loaded, the user can edit or define measurement positions. If the project already contains predefined positions, they will be displayed (as shown in Figure 3.1(b)). Otherwise, the image will appear without any markings, allowing the user to define new positions.

Positions provides several functionalities for working with measurement positions:

- Moving and zooming the image
  Press and hold the left mouse button, then drag the mouse.

  To zoom in or out, use the mouse wheel: Scroll forward to zoom in and scroll backward to zoom out. These operations enhance usability when defining measurement positions (Figure 3.2).

- Adding measurement positions
  To add positions, select the corresponding tool and follow these steps:

  1. Click on the "Add Points" icon in the toolbar.
  2. Move the cursor to the desired position on the image.
  3. Right-click to place a measurement point.

  Each point is marked with a circle, and a position number is automatically assigned. The assigned number, along with the X and Y coordinates, appears in the right panel.

> Positions are recorded in normalized coordinates.

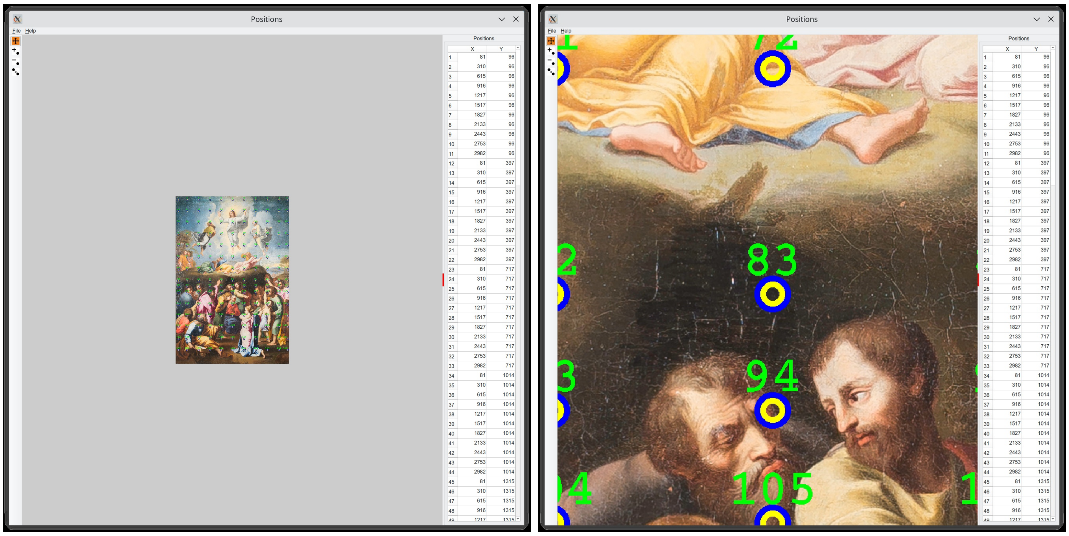


Figure 3.2: The mouse wheel allows to do zoom.

Users can add multiple positions as needed. During this process, image navigation (moving and zooming) remains available to facilitate accurate placement.

- Removing measurement positions
  To delete a position:

  1. Click on the "Remove Points" icon.
  2. Move the cursor to the position to be deleted.
  3. Right-click to remove the point.

  The corresponding circle and number will be erased from the image, and the position will be removed from the right panel.

  Positions must be deleted in reverse order (last added - first removed) to maintain sequential numbering and avoid renumbering complexities.

- Moving measurement positions
  To relocate a position:

  1. Click on the "Move Points" icon.
  2. Move the cursor near the position to be moved.
  3. Right-click to select the position (it will be highlighted).
  4. Move the cursor to the new location and right-click again to confirm the new position.

This allows users to adjust measurement positions dynamically by dragging them to different locations.

The right panel displays: Position order, X-coordinate and Y-coordinate. Additionally, it offers a quick search feature: Clicking on any cell in a positions row (order, X, or Y) highlights the entire row and the corresponding position on the image is marked with large circles for easy identification.

Once the positions have been marked and reviewed, the program provides two options for saving the information:

- Save image
  This option saves an image containing the marked positions in the results folder. The user can assign a custom filename; however, by default, the file will be named positions.png.
- Save positions
  This option saves the position data in the project_name.csv file. The following information is stored:
  - X and Y coordinates for each marked position.
  - Image width and height (in pixels).
  - Position numbers, recorded under the POSITION field.

  Below is an example of the resulting CSV entry:

```
XML;1.0.0
PROJECT_NAME;Test
AUTHOR;DMP
DATE;18/05/20
DEVICE;XRF;XRF pistol
TUBE;XRF;CU
XRF_DATA_ADJUSTMENT;ATOMIC_NUMBER
WIDTH_CM:50
HEIGHT_CM:50
WIDTH_PIXEL;1000
HEIGHT_PIXEL;1000
CS_ORIGIN;TOP_LEFT
POSITION;-;1;2;3
X;-;475;1273;1546
Y;-;2269;564;1369
```

> It is essential that the image used to define positions is identical to the one used by **XMapsLab** to ensure consistency in data alignment.

The Options menu allows users to customize how position marks are displayed, including settings for: Colors, font size and other display parameters (Figure 3.3).

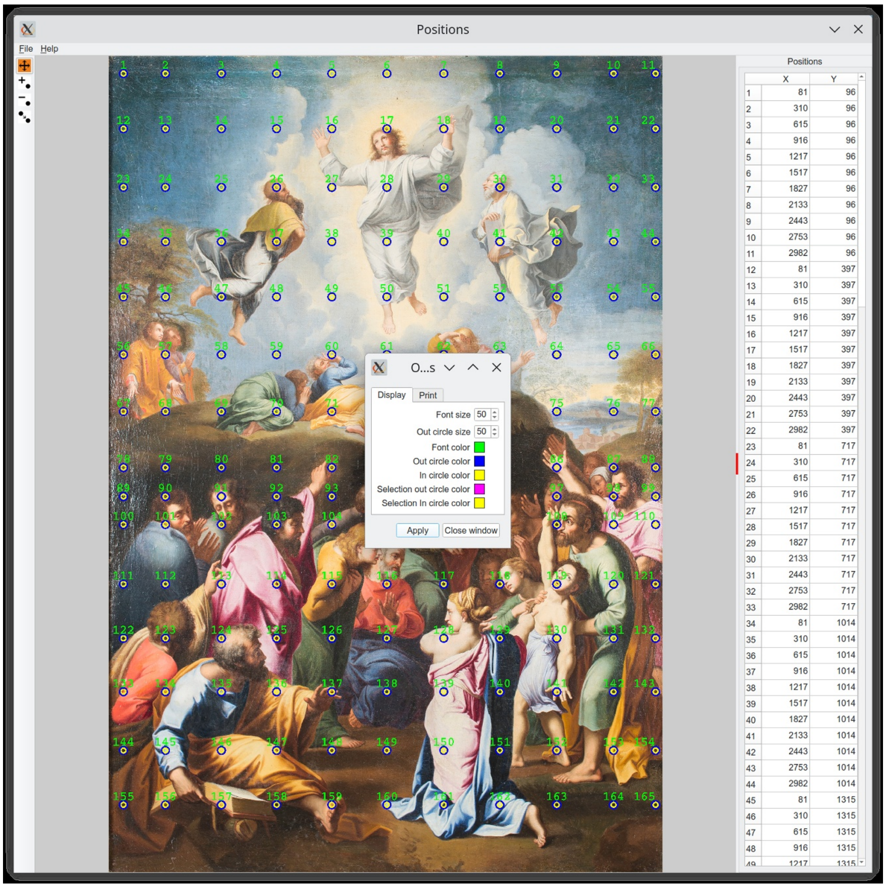


Figure 3.3: Widget to change the options

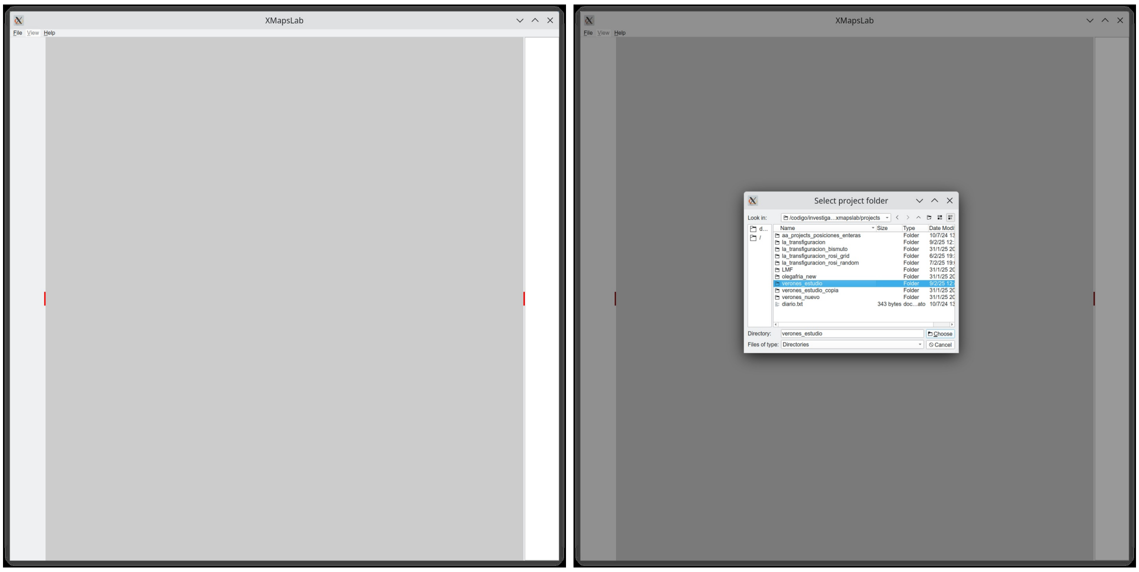
4

XMapsLab

XMapsLab is the program responsible for generating maps from the information stored in the project folder, including the image and the data contained in the project_name.csv file.

Before using **XMapsLab** a properly structured project must be created, as explained in the previous chapters. If the project is correctly set up, running **XMapsLab** is straightforward.

The general workflow consists of the following steps:



(a) Initial state

(b) Selecting a project folder

Figure 4.1: Two captures of the **XMapsLab** program in the initial state and selecting a project.

1. Load a project
2. Select the positions to be used for map generation
3. Choose the elements to be analyzed
4. Select an interpolation method
5. Generate the maps

Once the maps are created, **XMapsLab** provides tools for visualization and analysis. The following sections outline these steps in sequence.

1. Launching the program and loading a project
   Upon executing **XMapsLab** the main window appears, featuring a menu bar and three display areas.

   To load a project:

   (a) Click on File in the menu.
   (b) Select Load project (Figure 4.1(a)).
   (c) From the list of available projects, click on the desired project name.
   (d) Click Select to confirm the selection (Figure 4.1(b)).

   If the project contains multiple images, the program will prompt the user to select one. Likewise, if multiple data files are available, the program will request a selection.

   Once the project is loaded:

   - The color image is displayed in the central area.
   - The Positions tab is automatically selected (Figure 4.2(a)).

2. Selecting measurement positions for map generation
   By default, all measurement positions are selected for generating maps. However, users can modify this selection as follows:

   - To exclude a position, click on its corresponding row in the right panel. The position will be marked as invalid (Figure 4.2(b)).
   - To reselect a position, click on the row again. The position will become valid once more.

   If a project contains a large number of positions, manually enabling or disabling them individually can be time-consuming. To simplify this process, users can:

   - Toggle the status of all positions at once by clicking on the Valid column header.
   - Apply changes by clicking Update datathis ensures that updates take effect for map generation while preventing unnecessary multiple updates.

(a) Positions tab (b) Selecting positions

Figure 4.2: The selection of positions that will be used to create the maps.

3. Creating maps
   The next step is to generate maps based on the selected data. To do this, navigate to the Data tab (Figure 4.3(a)), where users can choose the elements and/or pigments for which maps will be created.

   There are two selection options:

   (a) Create maps for all available elements Click Create all maps.

   (b) Create maps for specific elements Select the desired elements by clicking on them, then click Create some maps (Figure 4.3(b)).

   Additionally, users can:

   - Add an identifier to the map name to distinguish different study conditions.
   - Select the interpolation method and adjust its parameters.
   - Modify the color palette, which determines how numerical values are represented visually.

4. Working with layers
   Once a map is generated, the program automatically switches to the Layers tab (Figure 4.4(a)). Each map is stored as a layer, and the layers are arranged in a depth order:

   - The color image layer appears at the top.
   - The most recently created map appears at the bottom.

(a) Data tab (b) Selecting some elements

Figure 4.3: The Data tab allows to select the element(s)/pigment(s) for which the map(s) will be created, the method to be used and its parameters.

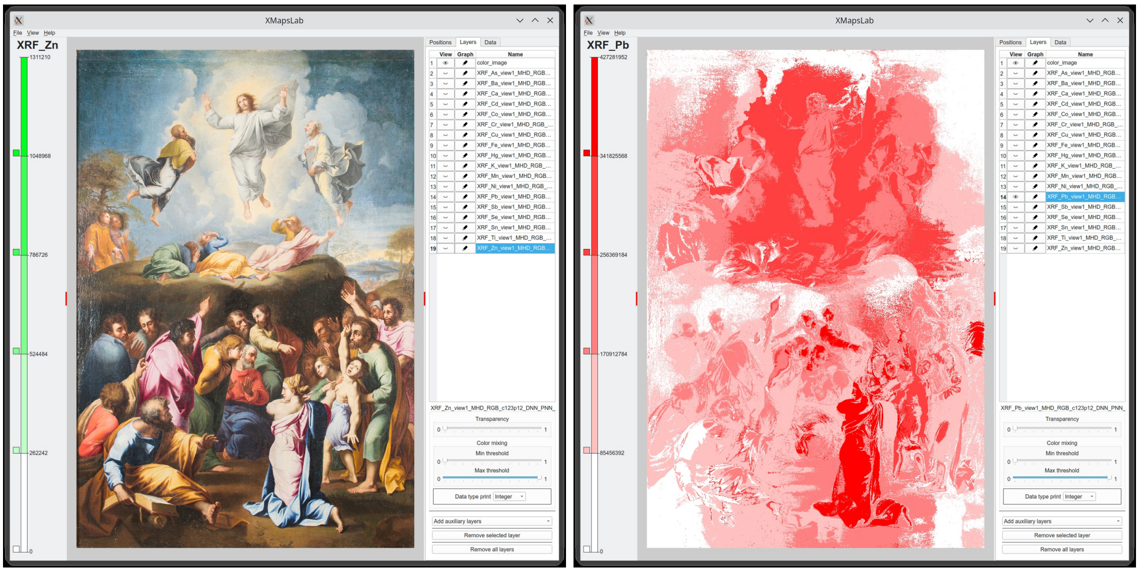
This depth ordering enables transparency and blending operations between layers.

After generating maps:

- The closest layer is automatically selected.
- The color bar of the selected layer appears on the left side of the interface.
  - The color bar visually represents the relationship between numerical values and colors.
  - The appearance of the color bar varies depending on the selected palette.
  - While the program assigns colors automatically, users can customize them.
  - The display format of the color bar can be adjusted as needed.

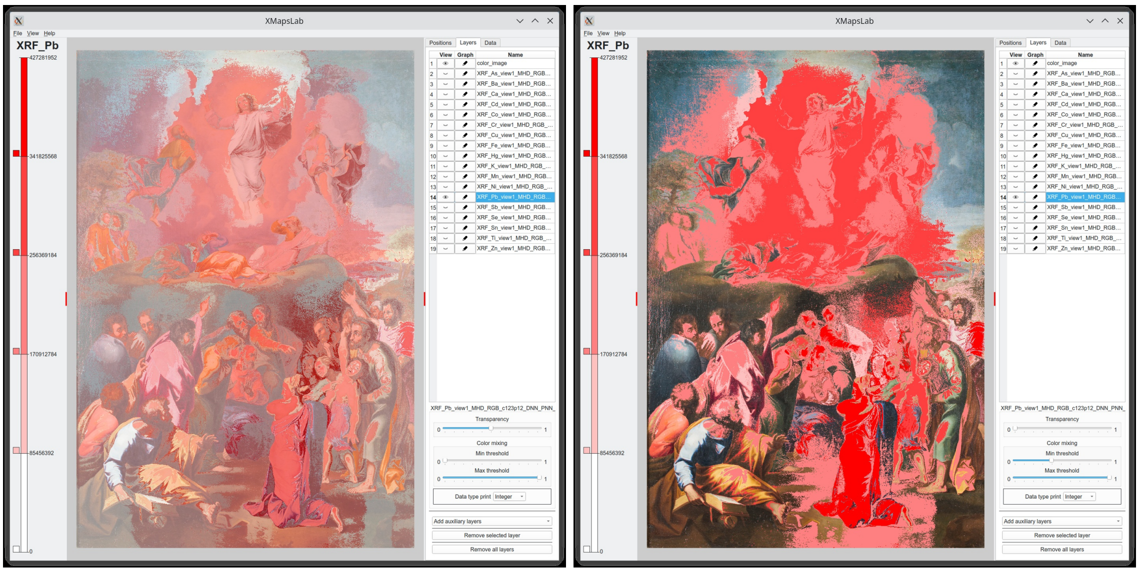
Each layer is assigned a unique name, which includes:

- The type data (XRF, XRD or RAMAN).
- The element/pigment name.
- A label
- The interpolation method used.
- The applied parameters.
- A layer creation order number, ensuring that even identical maps can be uniquely identified.



(a) Layers tab (b) Showing one map

Figure 4.4: The Layers tab allows you to select the displayed map and apply effects such as transparency and blending to it.



(a) Transparency (b) Mixing colors

Figure 4.5: It is possible to change the transparency and the mixing colors attributes of the maps.

This structured naming system allows users to manage and distinguish layers efficiently, even when multiple maps contain the same data.

To display a specific layer/map, click on the closed eye icon next to the desired layer. Once selected:

- The corresponding map is displayed.
- The layer is highlighted.
- The eye icon changes to an open eye (Figure 4.4(b)).

Users can select any layer, with support for displaying multiple layers simultaneously (up to 8 layers). However, layers follow a depth-based ordering, meaning:

- The topmost active layer is the most visible.
- If multiple layers are enabled, transparency and color mixing allow combining information from different maps.

Each layer includes adjustable settings for:

- Transparency
- Color mixing
- Displaying numerical values on the color bar

The Transparency slider controls the opacity of the selected layer:

- The default setting makes the layer fully opaque.
- Moving the slider to the right increases transparency, revealing lower layers.

If the next visible layer has no transparency, the selected layer's colors blend with it. However, if both layers have transparency, they will combine progressivelyup to a maximum of 8 layers.

> When more than 3 layers are active, distinguishing individual layer contributions may become challenging.

An example of transparency effects is shown in Figure 4.5(a).

In addition to transparency, color mixing offers another method for combining layers. This feature is controlled by two sliders:

- Minimum Threshold
  - Defines the lowest value at which the layer's data becomes visible.
  - Any values below this threshold are rendered fully transparent, allowing the next visible layer to show through.
- Maximum Threshold
  - Works in the opposite direction.
  - Values below this threshold remain visible, while values above it become fully transparent.

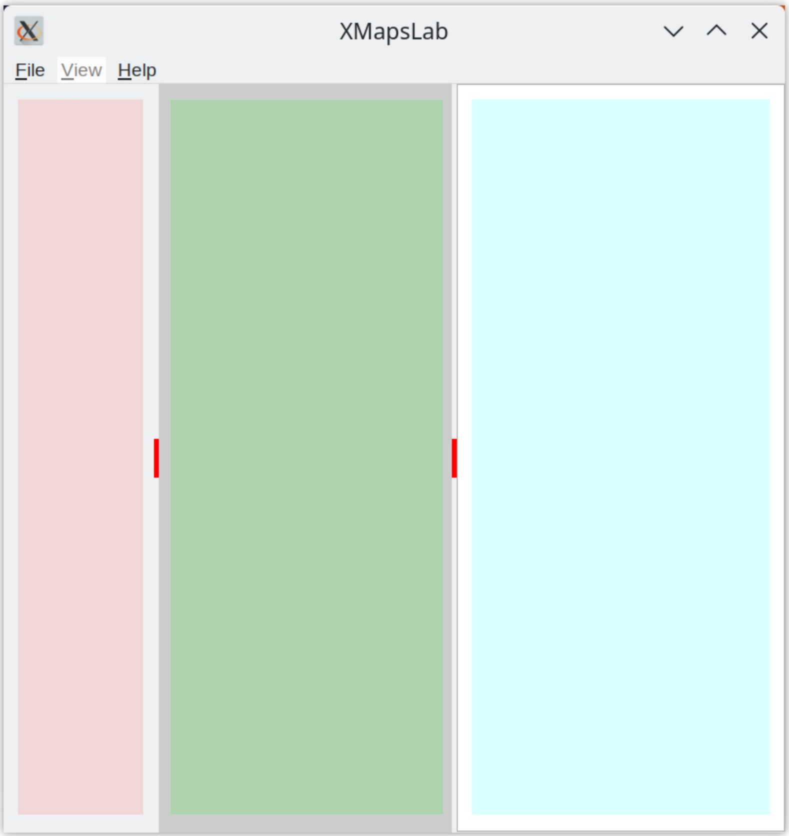


Figure 4.6: The main zones of the program: Color bar zone (red color), Display zone (green color) and Control zone (blue zone).

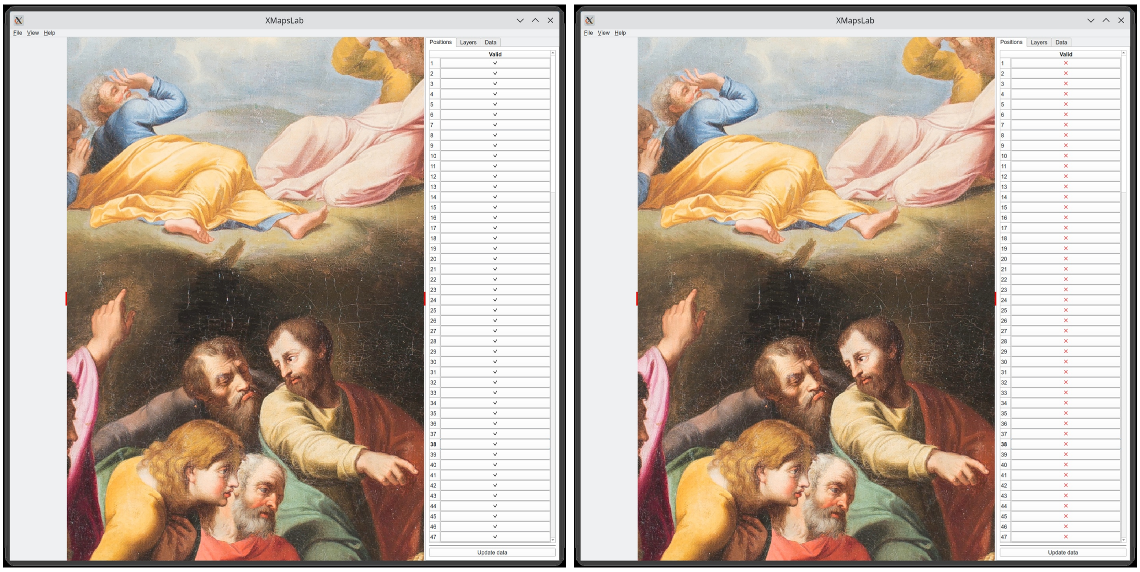
This selective filtering enables users to highlight specific value ranges within a layer. An example of color mixing effects is shown in Figure 4.5(b).

4.1 A detailed explanation

This section provides a detailed explanation of the different sections and functionalities of the **XMapsLab** program.

The interface is divided into three main sections, as illustrated in Figure 4.6. Each section serves a distinct purpose:

- Left Panel: Displays the color bar and associated numerical information for the selected layer. If the selected layer represents an element or compound map, the color bar provides a visual representation of numerical values. This section is particularly useful for interpreting the color-coding scheme used in the maps.



(a) List of positions (b) All positions deselected

Figure 4.7: The positions tab shows the list of positions and their state, selected or unselected. They can be changed individually or all at the same time.

- Central Display Area: Shows the resulting image, which can consist of:
  - A single layer
  - A combination of up to eight layers
  - Users can manipulate the image by adjusting the viewing angle and zoom level, allowing for detailed exploration of the data from different perspectives.
- Right Panel (Control Area): Contains the parameters and settings for calculations and display options, organized into multiple tabs, each dedicated to a specific functionality. Provides access to map generation, visualization, and customization options.

Understanding the layout and functionalities of these sections is essential for effectively using **XMapsLab**Ḃy familiarizing yourself with these areas, you can: Interact with maps dynamically, explore different visualizations with customized settings and adjust parameters to refine the analysis according to specific needs.

4.1.1 Control zone

In the right zone of the program, you will find tabs that enable you to access different types of actions and functionalities. Let's explore each tab in more detail:

- Positions tab
  Selecting the Positions tab displays a list of all measurement positions stored in the project_name.csv file (4.7(a)).
  - Each position is represented by a row with a button indicating its status (active/inactive).
  - Clicking the button toggles the position's state between active and inactive.
  - The status of all positions can be modified simultaneously by clicking the Valid column header (4.7(b)).
  - To apply the changes, click the Update data button.

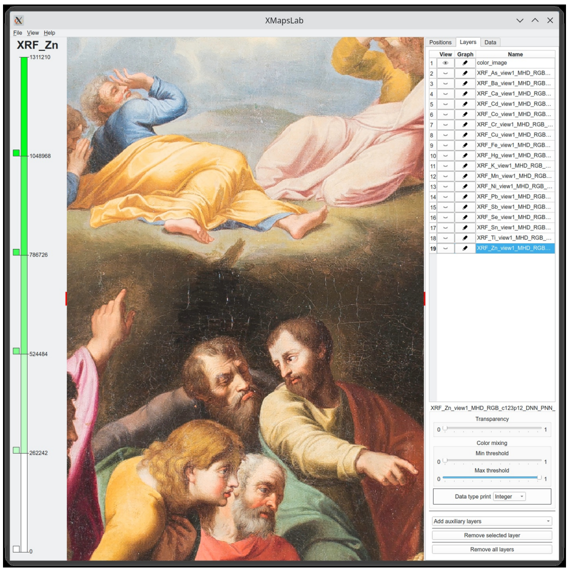


Figure 4.8: Once all the selected maps are created they appear as layers in a stack. The top layer is selected but is not visible. Its color bar is shown.

- Layers tab
  This tab displays all created layers, organized in a stacked structure (Figure 4.8).
  - Understanding layers and layers' stack
    * A layer represents a 2D image with associated parameters.
    * A layer stack is a collection of layers arranged in a depth-based order.
    * Layers are listed from top to bottom, but are displayed from bottom to top in the visualization area.

The initial image layer (background) is fixed and cannot be moved.

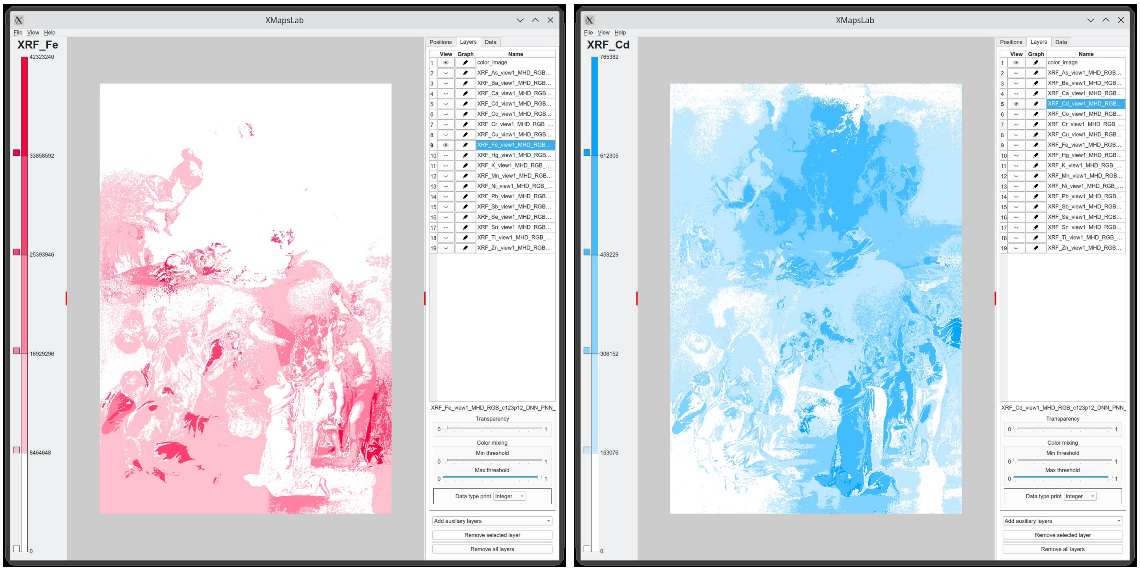


Figure 4.9: Making visible one layer. Its color bar is shown. Two examples with only one layer visible.

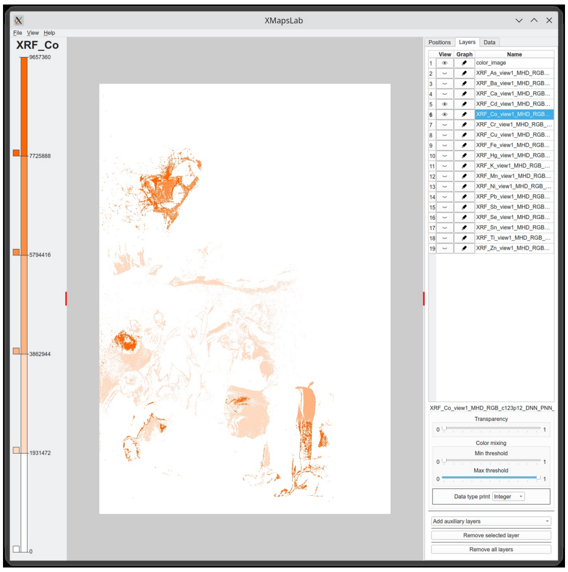


Figure 4.10: The order in the stack affects the final result. The top layer hiddens the one that is below.

- Modifying layer order

  * To rearrange layers, click on a layer, hold the left mouse button, and drag it up or down in the list.

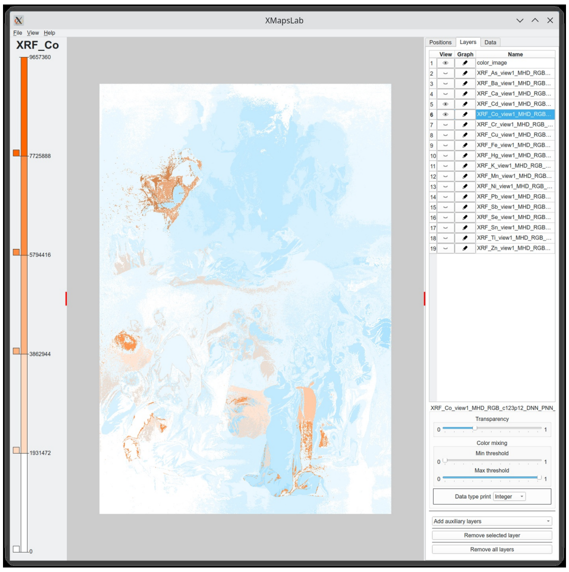


Figure 4.11: Applying transparency to the top layer allows to see the layer that is below.

 * The order of layers affects which layers are visible, especially when no transparency or color mixing is applied (Figure 4.10).

– Layer visibility and selection

 * Each layer has a visibility toggle, indicated by an eye icon (Figure 4.9):
 · Closed eye → Layer is invisible.
 · Open eye → Layer is visible.
 * If multiple layers are enabled, the topmost visible layer will dominate unless transparency or color mixing is applied.

– Adjusting transparency and color mixing

These operations apply to the currently selected layer.

 * Transparency adjustment
 · Controlled via a slider that modifies layer opacity.
 · Increasing transparency allows layers below to become partially visible.
 · If the top layer is fully opaque, lower layers remain hidden.
 * Color mixing (threshold-based substitution)
 · Minimum Threshold: Pixels below the threshold are replaced by corresponding pixels from the next visible layer. Pixels above the threshold remain unchanged.

· Maximum Threshold: Pixels below the threshold remain unchanged. Pixels above the threshold are replaced by pixels from the next visible layer.

This technique enables dynamic color blending, allowing for selective visualization of mapped values (Figure 4.12).

Transparency and color blending operations can be applied without any limitations.

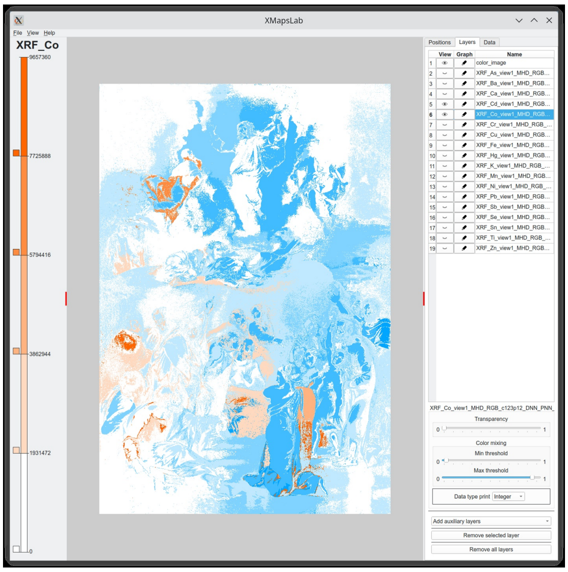


Figure 4.12: Applying color mixing to the top layer allows to combine it with the layer that is below.

* Type of numerical data
  Each layer also allows the adjustment of the type of numerical data displayed on the color bar. The available options are:
  · Integer
  · Real
  · Scientific notation
  · Percentage

The default display type can be modified in the Options menu.

* Auxiliary layers: Difference of Gaussians
  Additionally, the program supports auxiliary layers that are not directly associated with specific elements or compounds.
  Currently, the program offers the option to create a "Difference of Gaussians" (DoG) layer. This method generates an image that enhances contours

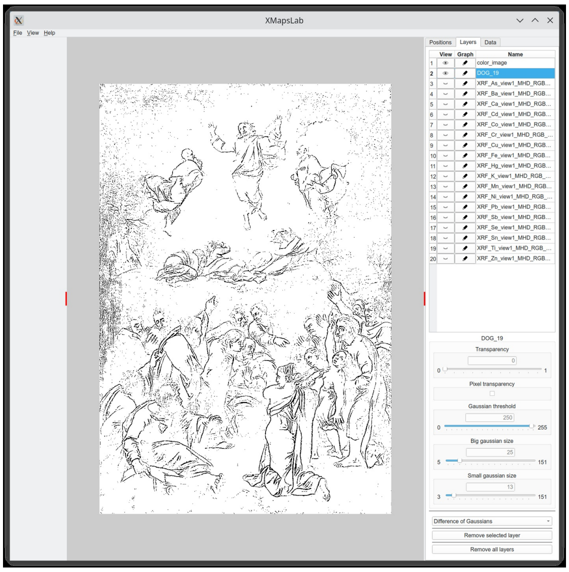


Figure 4.13: It is possible to create an image with the contours of the selected layer. The normal use is with the color image, which must be selected.

and boundaries, making it useful for identifying different regions or structural variations.
Like other layers, the DoG layer supports transparency adjustments for better integration with other data layers.

The primary parameters for adjusting the Difference of Gaussians method are:

- Gaussian threshold
- Big Gaussian size
- Small Gaussian size

Modifying these parameters alters the visual characteristics of the generated image, allowing users to fine-tune the contour enhancement for better analysis.

Typically, only the Gaussian size parameters require modification.

The Pixel transparency option is a distinct feature that differs from standard transparency settings. Instead of applying uniform transparency to the entire layer, it specifically targets contour pixels within the image:

- White pixels represent non-contour areas and become fully transparent.
- Black pixels represent contour areas and remain fully opaque.

This functionality allows for clear visualization of contours while maintaining transparency in non-contour regions.

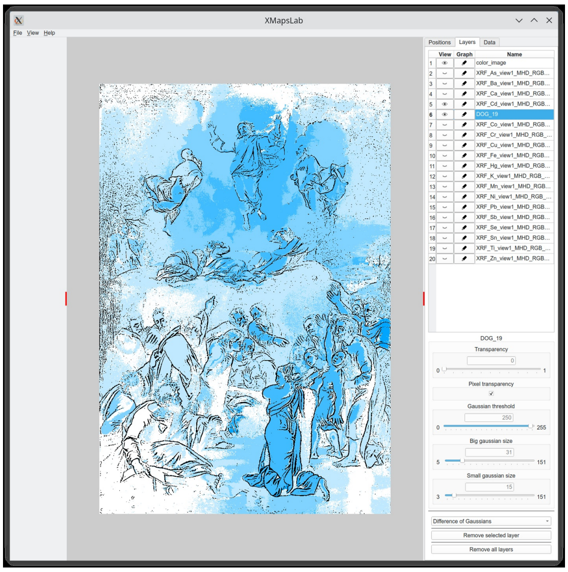


Figure 4.14: The layer with the contours image is moved above one map and the Pixel transparency is active to show the map.

When generating a contour layer using the “Difference of Gaussians” method, it is typically applied to the active layer. While it can be used on map layers, it is more commonly applied to the initial image layer (Figure 4.13).
The resulting contour layer is automatically placed above the selected layer in the layer stack. To visualize contours in relation to map layers, move the contours layer to the top of the stack. This ensures that contours remain visible while still displaying the underlying map data (Figure 4.14).

∗ Managing Layers
At the bottom of the Layers tab, several action buttons provide additional functionalities:
    · Remove selected layer: Deletes the currently selected layer.
    · Remove all layers: Deletes all layers simultaneously.

- Data tab
  The Data tab allows users to select the elements/pigments for interpolation operations.

  Users can choose which elements to include in the interpolation process:

  – All Elements → Click Create all maps to generate maps for all available elements.
  – Specific Elements → Click on the desired element(s) to select/deselect them, then click Create some maps.

⚠ New layers are placed above the selected layer in the stack. Initially, only the color image is present.

Figure 4.15: The parameters of MHD interpolation method.

If the objective is to generate maps for all elements, clicking Create all maps is more efficient than selecting elements manually.

When generating maps, users can customize the following parameters:

- View_name.
  This is a custom text that is added to the layer name. It allows to make easier to distinguish similar maps.
- Interpolation type.
  Specifies the interpolation function to be applied. Based on the selected method, additional customizable parameters become available.
  - Minimum Hypercube Distance
    For this method the following parameters can be changed (Figure 4.15):
    - Color 1 to 3 → Specifies which color components are used for distance calculations. For RGB, the mapping is: R=1, G=2, B=3.
    - Position 1 to 2 → Specifies which position coordinates are used for distance calculations. X-coordinate = 1, Y-coordinate = 2.
    - Distance normalization → Enables or disables distance normalization. When enabled, values are adjusted based on distance variations. When disabled, values remain constant despite distance variations.

      The original method applies normalization.

    - Position normalization → Controls whether X and Y positions are normalized. Can distort non-square images, affecting proportions.

Figure 4.16: The parameters of the Radial Basis Fuction interpolation method.

Figure 4.17: The possibility of changing the effect radius is shown with some basis functions.

The original method applies normalization.

- Probe → Defines the number of pixels used for calculating the color of each processed pixel. Options range from 1×1 (single pixel) to 49×49 (larger neighborhood). Larger values help reduce noise by averaging multiple pixels.

Figure 4.18: The parameters of the Triangulation interpolation method.

* Minimum 2D Cartesian Distance
  This method uses the same procedure as MHD but changing some parameters. The Voronoi diagram is obtained by unmarking the use of the color components and maintaining the position coordinates.
* RBF
  - Function → Selects the radial function used for interpolation. Options include Gaussian, Multiquadrics, Inverse Multiquadrics, etc (Figure 4.16).
  - Radius → Through this parameter, you can adjust the extension of the Gaussian, Multiquadrics, and Inverse Multiquadrics functions. Defines the spread of the selected radial function (Figure 4.17).
* Triangulation
  For this type we only have the possibility of adding or not the corners (Figure 4.18).
* Segmentation-based interpolation methods
  These methods use high-quality semantic segmentation to refine interpolation, SAM (Kirillov et al. [2023]):
  -
  - Segmentation-Based Minimum Hypercube Distance
    Applies the MHD method but restricts calculations to the segmented regions.
  - Segmentation-Based Mean
    Computes the mean of measurements within each segmented mask.
  - Segmentation-Based Minimum
    Computes the minimum measurement within each segmented mask.

  - Segmentation-Based Maximum
    Computes the maximum measurement within each segmented mask.
- Palette
  When generating maps, each calculated element value must be assigned a corresponding color based on a predefined color table, known as a palette.
  Different palette options are available, including:
  * Single-color palettes
  * Multi-color palettes
  * Continuous gradients
  * Sectioned gradients

  The default palette can be modified in the Options submenu to better suit specific visualization needs.

4.1.2 Display Zone

The display zone is used to visualize the resulting image after selecting, activating, and performing operations on the layers. It also provides additional information, including:

- Measurement positions
- Triangulation based on measurement positions (excluding and including corners)

User interaction and navigation:

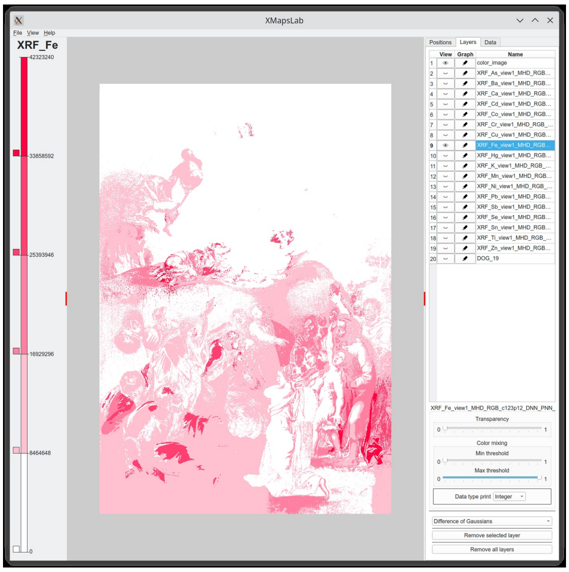
- Adjusting the observer's position → Click and drag with the left mouse button.
- Zooming in/out → Use the mouse wheel.
- Centering the camera → Double-click with the left mouse button.
- Resetting the camera to the initial distance → Double-click with the right mouse button.

4.1.3 Color Bar Zone

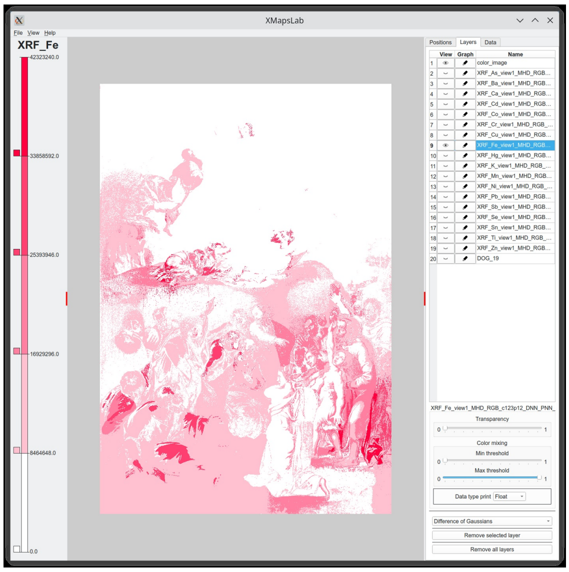
The color bar zone displays the color bar corresponding to the selected layer (see Figure 4.19) if the layer contains a map of an element or compound. Otherwise, the color bar remains empty.

The color bar visually represents the mapping of numerical values to colors, based on the selected palette in the XRF tab.

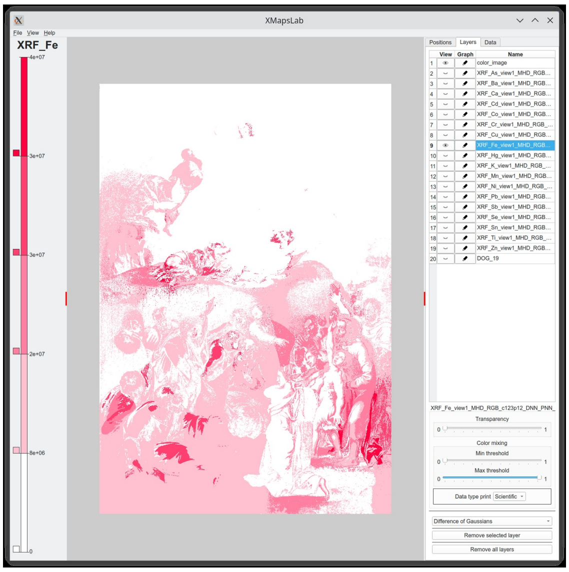
Numerical values on the color bar can be displayed in four formats:



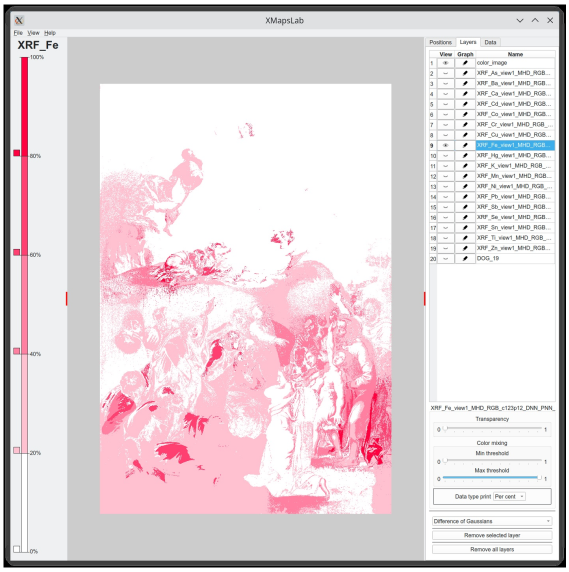
(a) Integers



(b) Reals



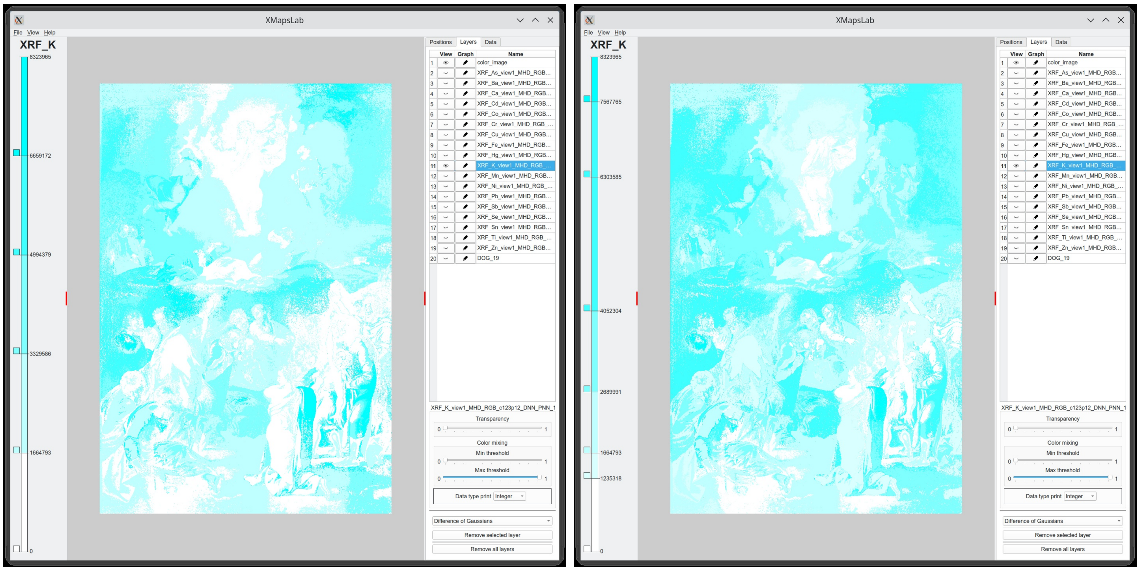
(c) Scientific



(d) Percentage

Figure 4.19: The color bar of the selected layer is shown in the left zone. The type of color bar depends on the selected palette. The numerical values can be shown in four types.

- Integer (Figure 4.20(a))
- Real numbers (Figure 4.20(b))
- Scientific notation (Figure 4.21(a))
- Percentage values (Figure 4.21(b))



(a) Original with integers (b) Using reals adding intermediante values

Figure 4.20: Different operations on the color bar that uses a discrete palette with 5 steps.

The number of steps selected in the palette determines how color ranges are assigned. The minimum number of steps is 2 (representing minimum and maximum values). Additional steps evenly distribute values across the range. Example: If 3 steps are chosen, the bar displays minimum, midpoint, and maximum values.

we distinguish between palettes that use tones and those that use colors.

- Tone-Based Palettes
  Use a single-color gradient, where the base color is assigned to the minimum value. The base color can be white or black and can be modified in the Options submenu.

  Two assignment methods are available:

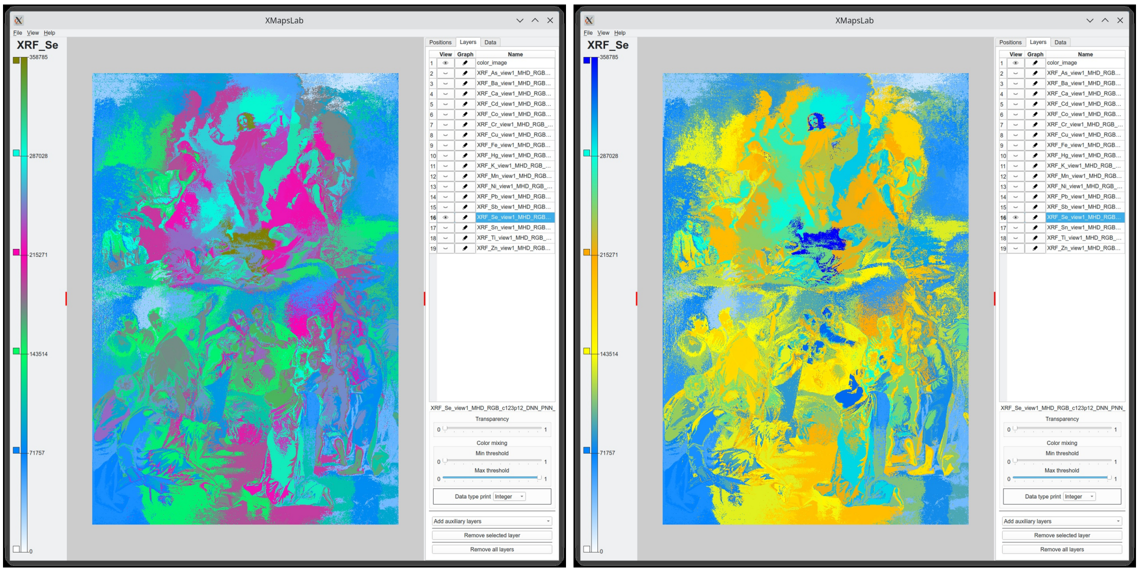
  – Gradual Assignment → Color transitions linearly between steps.
  – Step Assignment → Each color box represents a fixed value range and extends to the next step.

- Color-Based Palettes
  Similar to tone-based palettes, but with multiple colors instead of a single gradient. The same gradual and step-based assignments apply.

The color bar is customizable, allowing changes to individual color boxes:

- Changing the color of a box

  1. Double-click on the desired color box.



(a) Original (b) Changing the colors

Figure 4.21: Different operations on the color bar that uses a continuous palette with 5 steps.

2. A color selection window appears.
3. For shaded palettes, select the box at the upper end of the color bar.

- Adding a step/color box

  1. Double-click with the left mouse button at the desired position.
  2. The assigned value is automatically calculated.
  3. The color depends on the palette type: Tone palettes assigns the corresponding tone, Color palettes assigns a random color.

- Deleting a step/color box

  1. Double-click with the right mouse button on the box to delete it.
  2. The selected box is immediately removed.

- Moving a step/color box

  1. Left-click and hold on the color box.
  2. Drag it to the desired position.
  3. Release the mouse button to place the box.

(a) Positions (b) Position numbers

Figure 4.22: Auxiliary options to show positions and position numbers

(a) Standard triangulation (b) Triangulation with corners

Figure 4.23: Auxiliary options to show the triangulation

4.1.4 Menu options

The different menu options are listed below:

- File
  - Load project → Opens the selected project and loads all its data.
  - Close project → Closes the current project, allowing a new one to be loaded.
  - Save selected layer → Saves the map of the selected layer, with the option to include or exclude the color bar.
  - Save all layers → Saves maps for all layers in the project, with the option to include or exclude the color bar.
  - Save composed image → Saves the current view from the Display Zone, preserving transparency and color mixing operations.
  - Options → Opens a settings window for modifying various program parameters::
    * Display:
      · Font size → Adjusts the font size of text and circles marking positions in the display.
      · Customizes circle size, colors, and text size to ensure visibility across different images.
      · Measurement positions are centered within concentric circles.
    * Print → Adjusts the scaling factor for text printing, relative to the screen display size.
    * Layers → Defines the default numerical format for the color bar. Available options: Integer, Real, Scientific, Percentage.
- View
  - Show positions → Displays the measurement positions, aiding in the visualization of spatial distribution (4.22(a)).
  - Show position numbers → Displays the measurement position number if they are selected (4.22(b)).
  - Show triangulation → Displays the Delaunay triangulation of the measured positions, with two modes:
    * Normal → Displays triangulation excluding the corners (4.23(a)).
    * With corners → Displays triangulation including corners, where corners are assigned a value of 0 (4.23(b)).

4.2 View tools

In this section, we will present various analysis and visualization tools. The simplest tools are those that allow the visualization of position locations, the count of each position, and the composition of triangles obtained through Delaunay triangulation, both with and without corners, previously commented. The more advanced tools are those that enable the generation of statistical graphs regarding the distribution of elements in the positions selected by the expert, and particularly the **Laboratory** to which we will dedicate a separate section due to its importance.

The statistical visualization tools available are as follows: Bar chart, line chart and pie chart. The concept behind these tools is straightforward: they allow the display of statistical graphs representing the values of the selected layers, which correspond to elements and/or pigments, at the position chosen by the expert. That is, although only one layer (or a combination of layers, if transparency and color blending are used) can be viewed at a given moment, these tools enable the simultaneous visualization of the values of all selected layers for the chosen position, using one of the aforementioned chart types: bar, line, or pie. This provides information that would otherwise be difficult to display.

The selection of layers whose values are to be displayed simultaneously can be controlled independently of the layers currently activated for visualization. This is achieved using the pencil icon associated with each layer . By default, all layers are active, but they can be deactivated individually by clicking on the icon . Clicking the icon again will reactivate the layer. Since managing a large number of layers can be time-consuming and tedious, the state of all layers can be toggled to their complementary state by clicking on the header of the column labeled Graph. For example, if all layers are initially active, clicking on Graph will deactivate all layers. From there, specific layers can be reactivated by clicking on their respective pencil icons.

The first time the visualization of any of the statistical charts is activated, the maximum values of each selected layer will be displayed, with all values initially assigned the color black. Subsequently, once a valid position is selected, the corresponding values and colors will be shown. It should be noted that the maximum value displayed in the bar and line charts depends on the highest value among the maximum values of the active layers. In the case of pie charts, all values of the active layers at the selected position are summed to constitute 100%, thereby adjusting the proportions accordingly.

The accompanying images provide examples of the results obtained.

Another tool that can offer general information about a layer is the histogram. This tool displays the frequency distribution of the values within each layer. The purpose of the histogram is to present a graph indicating how many times each value occurs. Consider a simple example: a 2×2 pixel image with pixel values of 0, 0, 0.5, and 1. Here, the value 0 appears twice, while the other values appear only once. The resulting distribution list would be as follows: (0→2, 0.5→1, 1→1). If this list were displayed as a bar chart, the bar for the value 0 would be twice as tall as the bars for the other values.

Having understood the concept of a histogram, a question arises: How can this be applied to continuous variables? For instance, in an image with 6 million pixels, there could theoretically be 6 million unique values. How would such a large dataset be represented? A brief explanation is necessary to understand how this is possible when dealing with a continuous range of values stored in a floating-point variable. The key lies in recognizing that the range of values in a map, from the minimum (0) to the maximum value, must be mapped to colors. Although we can represent an infinite range of values, the number of colors available in computer representations is finite. Let us explore why.

Color representation in computers is based on the Red, Green, and Blue (RGB) color model. To create any color, the intensities of these three primary colors are combined. Think of it as having three flashlights: one emitting red light, another green, and the last blue. By combining their intensities, we produce the final color. A crucial aspect is the number of intensity levels available for each flashlight. In the simplest case, there are only two intensity levels: off or on. Here, we can represent $2 \times 2 \times 2 = 8$ distinct colors, with (0,0,0) representing black and (1,1,1) representing white (note that 0 corresponds to off and 1 to on). What happens if we have four intensity levels? For example, we could assign values from 0 (off) to 3 (maximum intensity), with 1 and 2 representing intermediate intensities of $1/3$ and $2/3$ of the maximum. In this case, the number of distinct colors would be $4 \times 4 \times 4 = 64$.

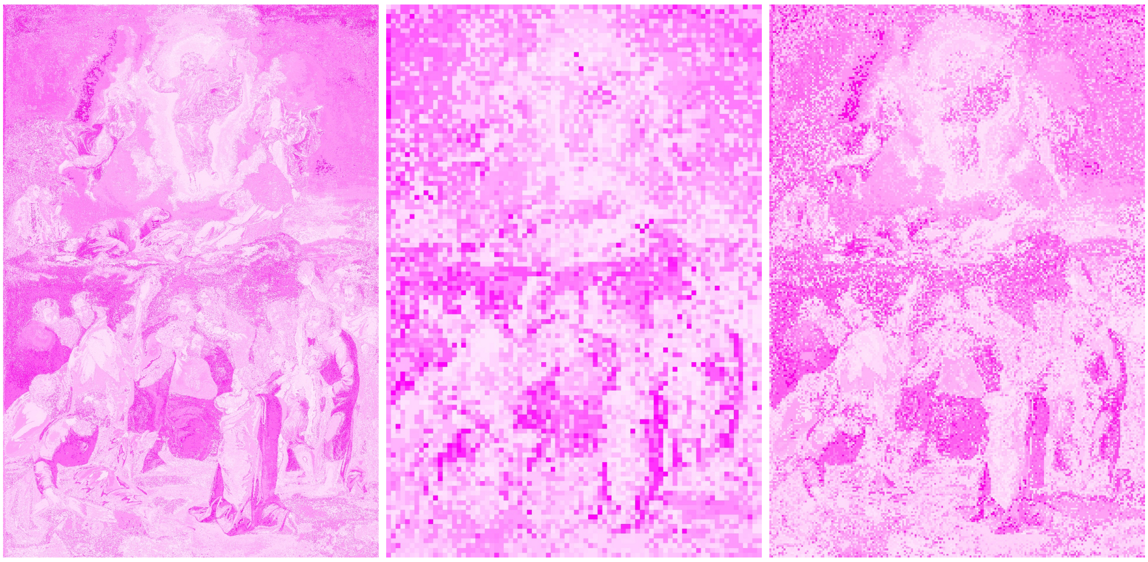
As we can see, each intensity level corresponds to a numerical value. It is now easy to understand why a color is represented as a triplet of numbers. The final piece of the puzzle is understanding how numbers are represented in computers. While one might assume that any number of intensity levels could be assigned to each component using real numbers, at the lowest level (despite the use of floating-point numbers in intermediate stages), an integer number of possibilities is assigned. Typically, each channel is allocated one byte (8 bits) to represent intensity levels. With 8 bits, we can represent $2^8 = 256$ possibilities, ranging from 0 to 255. Combining the three color components results in $256 \times 256 \times 256 = 16777216$ distinct colors.

One might assume that the range of intensity values must therefore correspond to the range of colors, from 0 to 16.7 million. However, this is not the case, as we aim to represent changes in intensity through a color gradient rather than a random assortment of colors. To achieve this, the range of intensity values is mapped to 256 possibilities. For example, if we wish to create a gradient of red, we can vary the red component across its 256 levels while keeping the green and blue components at 0. Thus, the range of values is divided into 256 intervals, and each value is counted within one of these intervals. The result is a graph that indicates how the values are distributedwhether certain ranges are more densely populated, whether the distribution is uniform, and so on. The histogram is calculated for each layer.

4.3 Using spot size

XMapsLab can work with measurements taken from different devices. The only requirement is that all measurements must be performed at the same positions; otherwise, the maps will not align perfectly.

One of the characteristics of different devices is the area in which the measurement is taken, known as the spot size. This area is a circle whose diameter is specified. It is clear that a device with a larger spot size has a lower ability to distinguish fine details in the artwork. To account for this characteristic, **XMapsLab** provides the option to apply the necessary



(a) Virtual spot size (b) Spot size 8 mm (c) Spot size 3 mm

Figure 4.24: Example to illustrate the effect of the spot size.

adjustments to consider the spot size. By default, the program uses a spot size corresponding to the size of a single pixel in the color image, which generally represents a much higher sampling density than that of the actual device.

To apply the effect of the spot size, the program adjusts the image size and calculations according to the dimensions obtained based on the spot size. For example, suppose we have a square image measuring 100×100 cm, with a corresponding resolution of 1000×1000 pixels. The virtual spot size would be $1000/1000 = 1$ mm. If the device's spot size were 10 mm, the image would need to be resized to 100×100 pixels, where each pixel represents 10 mm. It is important to note that there is no exact match between the area of a circle and the area of a square.

Figure Figure 4.24 illustrates an example of the results that can be obtained with different spot size values.

4.4 Laboratory

The **Laboratory** module is designed to represent hypotheses about pigment presence by analyzing the spatial distribution and intensity of elements. This is accomplished through a set of operations tailored to each specific hypothesis. For instance, identifying potential cinnabar regions requires detecting both Hg and S simultaneously, generating a new map that highlights these areas. The process begins by converting continuous values into binary form using a predefined threshold. These binary values are then combined using logical operations. In the case of cinnabar detection, an AND operation is applied, ensuring that a positive result is only returned when both input values are true; otherwise, the output remains false. Beyond binary operations, continuous value-based computations allow for a more refined approach to pigment mapping. For example, calculating the geometric mean of Hg and S intensities ($\sqrt{Hg * S}$) can produce a more nuanced map indicating possible cinnabar presence.

A particularly useful feature within this analytical framework is the ability to identify specific colors within an image. This function operates similarly to the previously described methods but with a focus on color, which directly relates to pigment composition. Since the images serve as element distribution maps, they can be combined with elemental data to create composite maps. The goal is to isolate a specific color while distinguishing it from others. However, relying on the RGB model proves inadequate for this purpose, as it effectively differentiates primary colors (red, green, and blue) but struggles with mixed hues. For instance, while it easily recognizes pure red (1,0,0), it lacks precision when differentiating shades like yellow (1,1,0) or orange (1, 0.6, 0.2). To overcome this limitation, a more suitable color model is needed. In this case, the HLS model (Hue, Lightness, Saturation) is used, as it allows for precise hue-based differentiation while minimizing the effects of brightness (black component) and saturation variations (white component). With this foundation in place, the following sections provide a detailed explanation of the applied techniques.

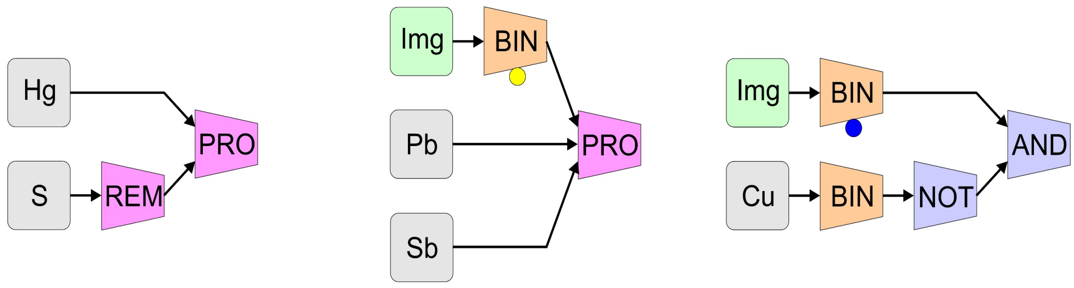
To enable the definition of diverse combinations, a specialized functional language has been developed. This language follows a structure where a set of inputs undergoes a defined transformation, yielding an output. Functions can take one or multiple inputs to process data, whereas others generate results without requiring input values. This design enables the construction of complex relationships through the strategic combination of functions. These defined functions operate at the pixel level across maps or images, ultimately producing a new output map.

This language categorizes functions into five types:

- **Image function:** This function represents the RGB image and only has an output.
- **Element functions:** These functions represent the element maps that are produced with the interpolation methods and only have an output.
- **Conversion functions:** These functions produce a conversion of the input data to generate the output data. Two conversion functions have been defined:
  - **Binary (BIN):** Converts continuous values to a binary value: true or false. Below a threshold (selected by the user), all values become 0, and above it, they become 1, implying non-existence or existence, respectively. It takes one continuous input and produces one binary output.
  - **Color to binary (COL BIN):** This function allows searching for a specific color (selected by the user) in the original image, with a certain level of tolerance, and obtaining a binary map. It takes the color image as input and produces a binary output.

- **Logical functions:** These functions have two inputs, which must be binary, and produce one binary output. The implemented logical functions are as follows:
  - **NOT (~):** Inverts the input; 0 becomes 1 and 1 becomes 0.
  - **AND (&):** Performs a logic AND operation. This means that a 1 is obtained only if both inputs are 1. Otherwise, the result is 0.
  - **OR (|):** Performs a logic OR operation. This means a 1 is obtained if both inputs are 1 or if either of them is one. Only if both are 0, the result is 0.
- **Numeric operations:** These functions enable operations on two elements with real values, producing real values. The function incorporates a third input, which must be a binary map, producing the effect that the function is applied only where the logical map is true: If the binary map value is true, the operation is applied and a value is produced. Otherwise the result is 0. By default, if the third input is not included, all positions are considered true. The following operations have been implemented:
  - **Element:** The result is the value of the element only where the binary map is true. For example, this function can be used to get the map where one element is present, depending on a color.
  - **Element Remove:** The result is the value of the element removing some threshold only where the binary map is true. For example, this function can be used to get part of an element, which can be useful when it is combined in several pigments.
  - **Product:** Applies the geometric mean. Given the A and B inputs it produces $\sqrt{A*B}$ as output. For example, this function can be used to get the map of a relation between two elements.
  - **Sum:** Applies the operation A+B.
  - **Difference:** Applies the operation ABS(A-B) (ABS means absolute value). For example, this function can be used to get the map with the difference of two maps produced with different interpolation methods.
  - **Percentage Sum:** Applies the operation A%+B%. This means it is performed if the values of A and B meet certain proportions. For example, to look for areas where A is twice as much as B. This can be used to simulate a chemical formula.

The set of implemented functions, designed for both independent and combined use, enables the empirical evaluation of hypotheses or specific conditions within the analytical framework. For example, determining the presence or absence of a particular element in a map involves applying a binary conversion operation followed by negation, allowing for the identification of regions where the element is missing. Likewise, to confirm the coexistence of two elements, an AND operation or multiplication can be applied, effectively isolating areas where both elements are present. Expanding on this principle, the presence of three distinct elements can be verified through the sequential application of two AND or multiplication operations, facilitating a more comprehensive analysis of elemental distribution.



(a) Finding cinnabar (b) Lead (Pb) and antimony (Sb) in yellow zones (c) No copper in blue zones

Figure 4.25: Block diagrams of the various example operations. (4.25(a)) The maps of mercury (Hg) and sulfur (S) removing a threshold is firstly obtained, and then are combined with an PRODUCT operation. (4.25(b)) A binary map of the zones with yellow color is produced with the COLOR_BIN operation. A PRODUCT operation is used to combine the lead and the antimony, but controlled by the color map. (4.25(c)) A binary map of the zones with blue color is produced with the COLOR_BIN operation. A binary map of the copper is obtained with the BINARY operation. Then the result is negated with the NOT operation. Finally, the color map and the NOT map are combined with an AND operation.

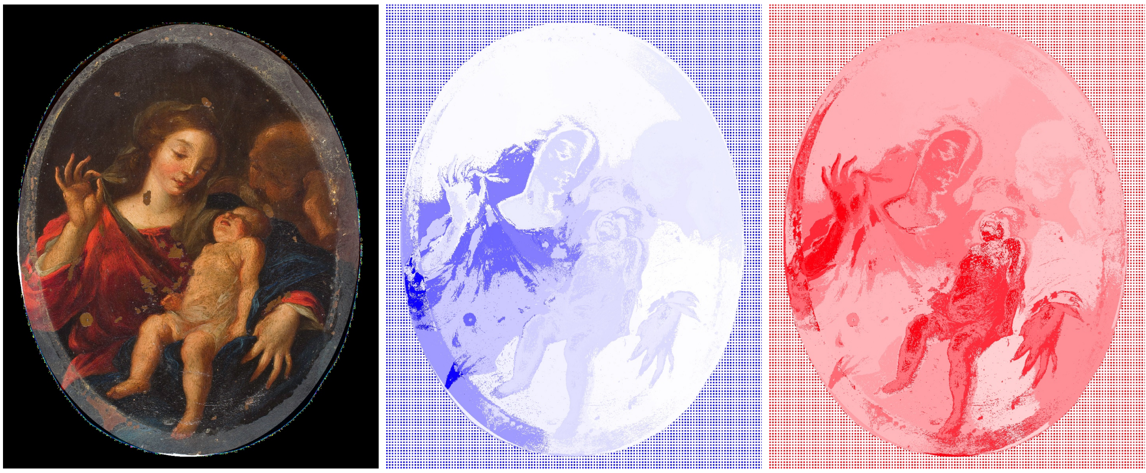
A key aspect of this system is the integration of the functional language into a graphical block editor. This editor represents functions as visual blocks and their interconnections as linking lines, illustrating the relationships between operations. Through an intuitive drag-and-drop interface, users can construct custom function sequences effortlessly, streamlining the assembly of analytical workflows. This method not only enhances usability but also supports the iterative refinement of complex analyses. Additionally, the editor maintains a record of all operations, allowing users to trace their analytical process and fully understand how final results are derived.

The efficiency and user-friendly nature of the block editor are demonstrated in an example video available on the project's website, showcasing the capabilities of the **Laboratory** module.

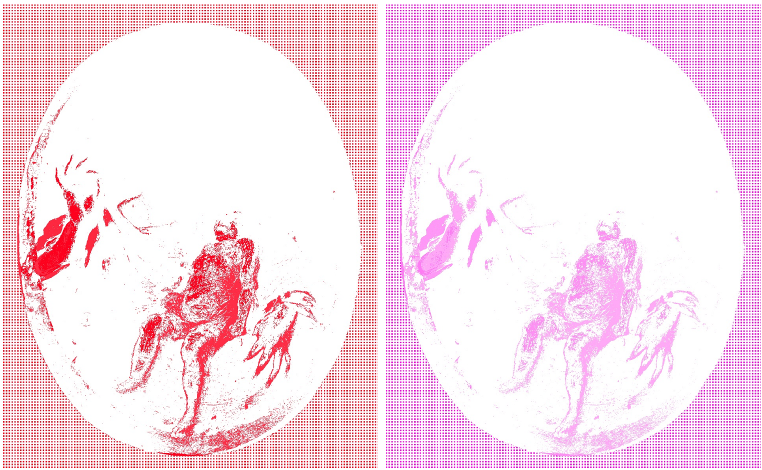
4.4.1 Illustrative examples of Laboratory

The Laboratory's versatility is highlighted through three exemplary cases, demonstrating its capacity to facilitate pigment analysis in artworks:

Cinnabar identification: This hypothesis can be tested in several ways. The simplest is to obtain the binary maps where there are mercury and sulfur and combine them with an AND operation. Another possibility would be to use the PRODUCT operator. As has been proven by checking the maps, there is a gypsum base, part of the sulfur combines with calcium and another part with mercury. To check this we can eliminate part of the sulfur with the



(a) Color image (b) Mercury (Hg) map (c) Sulfur (S) map



(d) The zones where there is sulfur after removing 70% (e) The zones where both elements are present

Figure 4.26: Operating with maps: The possible presence of cinnabar can be verified by combining the areas where there is mercury (Hg) and sulfur (S). In this example, we have supposed that 70% of sulfur was combined with calcium (gypsum), leaving 30% to combine with mercury. The area with a grid indicates that there are no valid values.

REMOVE operation, using the rest to combine with the mercury using the PRODUCT operation. Since the results are obtained in real time, the user can check the result by varying the different parameters until the searched result is obtained. The resultant maps, illustrating the areas where cinnabar might be present, are depicted in Figure 4.26, with the corresponding analytical process outlined in 4.25(a).

Naples yellow detection: Identifying Naples yellow involves isolating regions displaying a yellow hue and assessing the co-occurrence of lead (Pb) and antimony (Sb). This hypothesis can be verified by obtaining the area where the yellow color is found (with a certain level of tolerance), which controls where the PRODUCT of lead and antimony is applied. This combination generates maps that delineate areas potentially containing Naples yellow. The example images are found as additional material. The block diagram in shown in 4.25(b).

Exclusion of copper-based pigments: The absence of copper (Cu) in areas predominantly blue or green can exclude the presence of pigments such as azurite, Egyptian blue, or malachite. This hypothesis can be verified by obtaining the area where the color blue is found (with a certain level of tolerance), combining it with the map where copper is not found, which is obtained first by obtaining the binary version and then applying the NOT operator. The example images are found as additional material. The block diagram in shown in 4.25(c).

These examples underscore the Laboratory module's capability to support complex analyses, enabling experts to explore various hypotheses concerning pigment composition and distribution with precision and efficiency. Through iterative operations and the integration of color-based analyses, the module offers a dynamic platform for the detailed study of artworks.

5

Interpolation methods

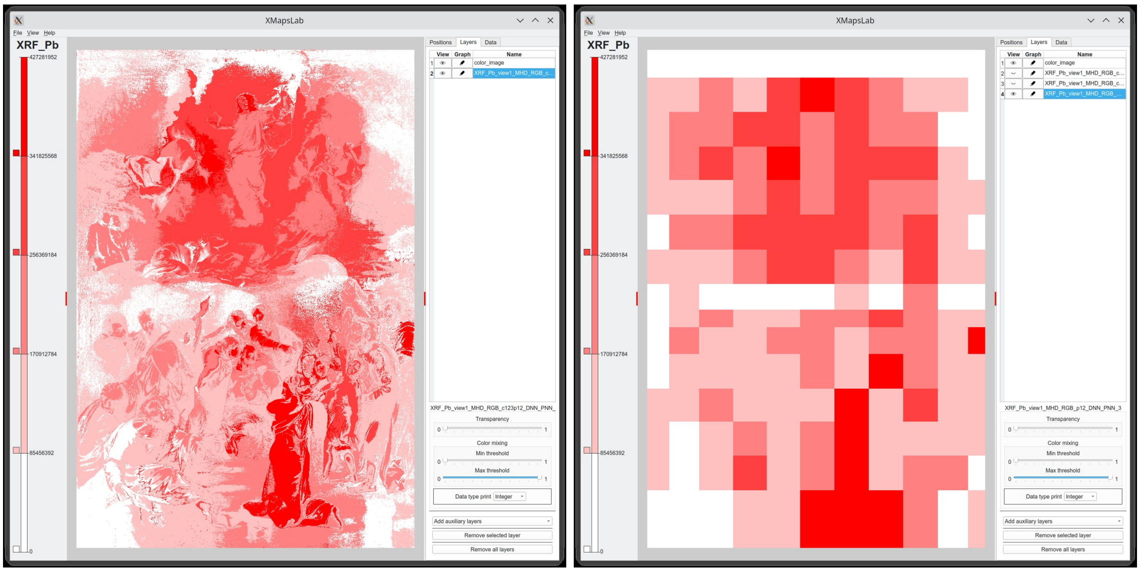
Among the various methods available for interpolating spatial data, we have implemented several approaches capable of handling scattered positions rather than a grid. These methods can also produce values outside the convex hull (the smallest convex polygon that includes all measured points [Graham and Frances Yao, 1983]).

XMapsLab includes four interpolation methods:

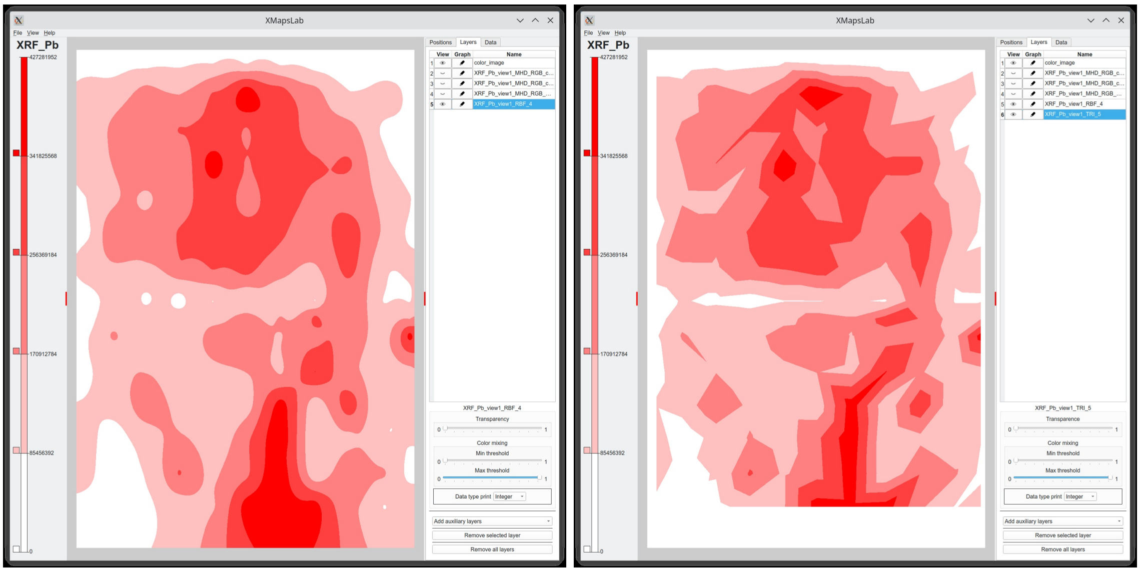
- Minimum Hypercube Distance (5.2(a))
  This method, developed by Martín-Ramos and Chiari [2019], utilizes both normalized RGB color values (R,G,B) (where R,G,B range from 0 to 1) and normalized position coordinates (x,y) (where x,y range from 0 to 1) to calculate distance. Each non-measured position has five known parameters (RGB and position coordinates) and one unknown parameter (the value, $(R,G,B,x,y,?)$). The method computes the distance between the unknown position and all known positions using the formula: $d = \sqrt{(R-R_i)^2+(G-G_i)^2+(B-B_i)^2+(x-x_i)^2+(y-y_i)^2}$. The value of the nearest known measurement (i.e., the one producing the smallest distance) is assigned to the non-measured position. Additionally, a modulation factor can be applied by computing the ratio between the minimum computed distance and the maximum distance within the hypercube: $f = \frac{Max_{distance-d}}{Max_{distance}}$. Since the method described by Martín-Ramos and Chiari [2019] does not have an assigned name, we refer to it as Minimum Hypercube Distance (MHD) (Martín et al. [2023]).

- Minimum 2D cartesian distance (5.2(b))
  Another useful interpolation method is based on the Voronoi diagram [Aurenhammer, 1991]. This approach classifies unknown positions according to nearest-distance criteria, given a set of n seed positions.

  - The distance between each unknown position and each seed is computed.
  - The value of the unknown position is assigned to that of the nearest seed.



(a) Minimum Hypercube Distance (b) Minimum 2D cartesian distance

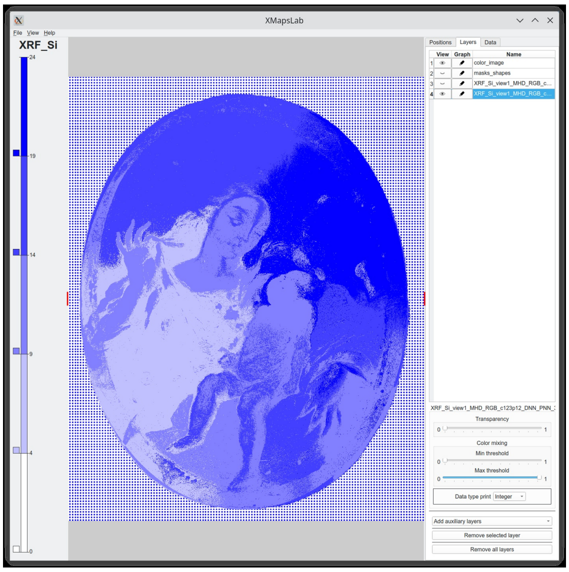


(c) Radial Basis Function (Linear) (d) Triangulation (without corners)

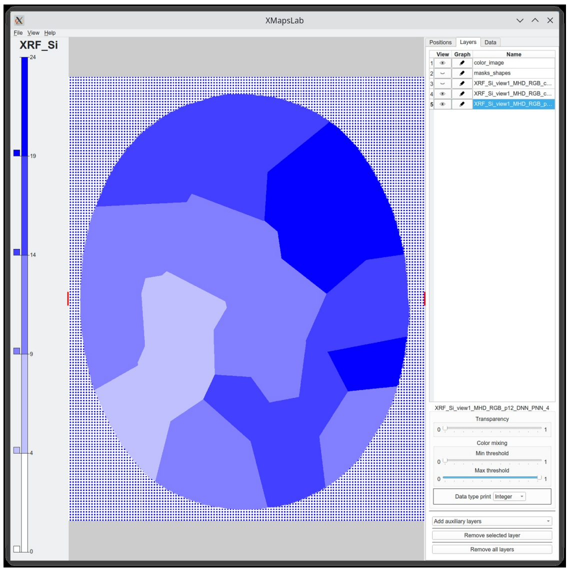
Figure 5.1: Results of the four interpolation methods using the same data with example A.

– Any additional values associated with the seeds are transferred to the unknown positions.

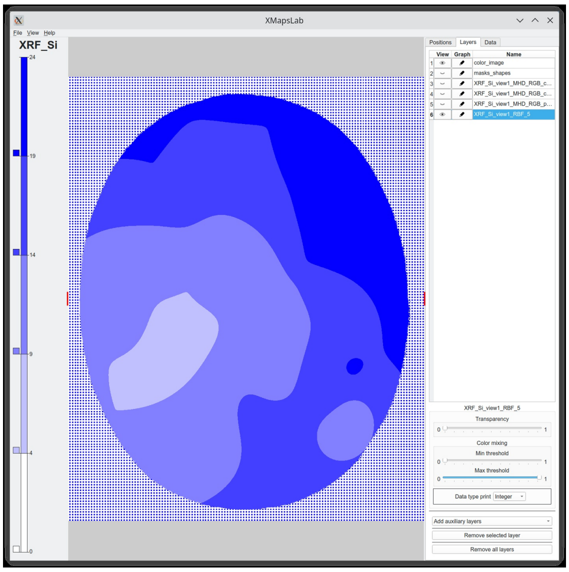
In this case, the measured values are transmitted and then mapped to colors. From this perspective, it is evident that the Minimum Hypercube Distance (MHD) method extends the Voronoi diagram into five dimensions, incorporating color into the position coordinates while transmitting the measured values.



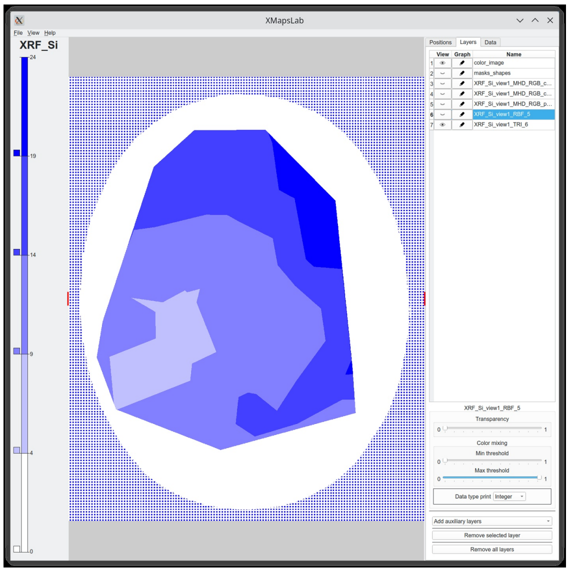
(a) Minimum Hypercube Distance



(b) Minimum 2D cartesian distance



(c) Radial Basis Function (Linear)



(d) Triangulation (without corners)

Figure 5.2: Results of the four interpolation methods using the same data with example B.

In terms of visualization, this method can be imagined as a simulation where different colored paints are poured into each seed position and spread outward until they collide with other colors. A gradual transition effect can also be simulated as the paint spreads.

We have included this method as a simplified version of MHD, making it a useful complement to the other available interpolation techniques.

- Radial Basis Fuction (5.2(c))
  Interpolation using Radial Basis Functions (RBFs) involves constructing an interpolant by computing a weighted sum of RBFs. A radial basis function, as defined by Hardy [1971], is a function that depends on the distance from a fixed point. For example, a Gaussian function satisfies this definition.

  The basic approach is as follows:

  - A radial function is placed at each known point.
  - The weights of the functions are adjusted.
  - The weighted functions are combined to form the interpolant.

  The ε parameter for Gaussian or multiquadric functions is set by default to the average distance between known points. Additionally, a smoothness parameter allows for controlling the smoothness of the approximation. By default, interpolation is performed at the known points.

  Our implementation includes support for multiple radial basis functions, providing different behaviors and characteristics for greater flexibility in the interpolation process. We have tested the following functions:

  - Multiquadric
  - Inverse Multiquadric
  - Gaussian
  - Linear
  - Cubic
  - Quintic
  - Thin Plate Spline

- Triangulation (5.2(d))
  We have implemented an interpolation method using normalized barycentric coordinates with triangles. This coordinate system, introduced by Möbius and Barth [1827], enables the computation of values at any position within a triangle, given at least three measurements (V_1, V_2, V_3) at three distinct positions: $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, and $P_3(x_3, y_3)$. The absolute barycentric coordinates (α, β, γ) of the triangle are used in this computation.

  The absolute barycentric coordinates satisfy the following conditions:

$$\alpha + \beta + \gamma = 1$$
$$0 \leq \alpha, \beta, \gamma \leq 1$$

  To compute the position within the triangle, we use the following formulas:

$$x = \alpha x_1 + \beta x_2 + \gamma x_3$$
$$y = \alpha y_1 + \beta y_2 + \gamma y_3$$

These formulas allow us to interpolate the value at any position within the triangle based on the known measurements at the three vertices. For example, if the values at the three positions were $(1000, 500, 600)$, then the value at the center of the triangle would be $\frac{1}{3} \times 1000 + \frac{1}{3} \times 500 + \frac{1}{3} \times 600 = 700$.

To apply this interpolation method to a set of samples, we first convert the positions into a set of triangles using the Delaunay triangulation method [Delaunay, 1934]. This method ensures that the generated triangles possess desirable properties, such as maximizing the minimum angle and avoiding sliver triangles.

It is important to note that if corner samples are missing, the resulting triangulation may not cover the entire image. In such cases, artificial corners with a value of 0 can be added to ensure full coverage.

- Segmentation-based methods
  Currently under review.

6

How to use CSV files with a spreadsheet

In this chapter, we will guide you through the process of using a spreadsheet to create or edit CSV files. While most spreadsheet applications support this functionality, we will focus on LibreOffice, a widely used spreadsheet program that is compatible with Excel and available for various operating systems, including Windows, Linux, and macOS.

If LibreOffice is not installed on your system, you can visit its official website by searching for “libreoffice.org”. From there, select the download option and follow the installation instructions.

Once LibreOffice is installed, you can follow the steps below to create a CSV file:

1. Launch the LibreOffice program. Upon opening, you will see the initial state of the application, as shown in Figure 6.1. A blank spreadsheet will open, allowing you to enter your data.

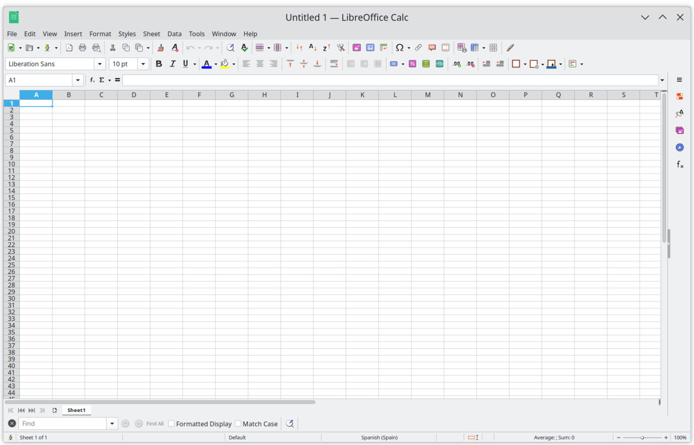
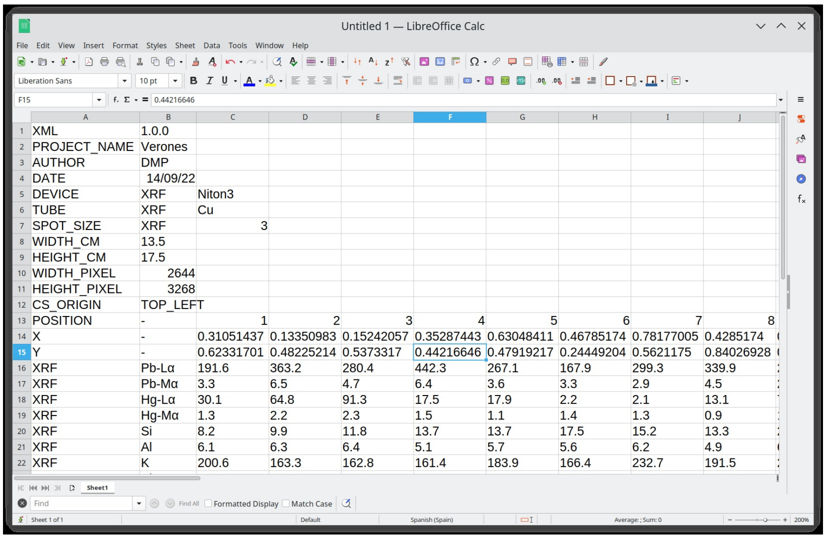


Figure 6.1: Initial state of LibreOffice

2. Enter the data. Fill in the cells with the desired values and organize them into columns and rows according to your requirements. We will introduce the data of a very simple project. The meaning of each field was explained in chapter 2, page 15. The result is shown in Figure 6.2.



	A	B	C	D	E	F	G	H	I	J
1	XML	1.0.0								
2	PROJECT_NAME	Verones								
3	AUTHOR	DMP								
4	DATE	14/09/22								
5	DEVICE	XRF	Niton3							
6	TUBE	XRF	Cu							
7	SPOT_SIZE	XRF	3							
8	WIDTH_CM	13.5								
9	HEIGHT_CM	17.5								
10	WIDTH_PIXEL	2644								
11	HEIGHT_PIXEL	3268								
12	CS_ORIGIN	TOP_LEFT								
13	POSITION	-	1	2	3	4	5	6	7	8
14	X	-	0.31051437	0.13350983	0.15242057	0.35287443	0.63048411	0.46785174	0.78177005	0.4285174
15	Y	-	0.62331701	0.48225214	0.5373317	0.44216646	0.47919217	0.24449204	0.5621175	0.84026928
16	XRF	Pb-Lα	191.6	363.2	280.4	442.3	267.1	167.9	299.3	339.9
17	XRF	Pb-Mα	3.3	6.5	4.7	6.4	3.6	3.3	2.9	4.5
18	XRF	Hg-Lα	30.1	64.8	91.3	17.5	17.9	2.2	2.1	13.1
19	XRF	Hg-Mα	1.3	2.2	2.3	1.5	1.1	1.4	1.3	0.9
20	XRF	Si	8.2	9.9	11.8	13.7	13.7	17.5	15.2	13.3
21	XRF	Al	6.1	6.3	6.4	5.1	5.7	5.6	6.2	4.9
22	XRF	K	200.6	163.3	162.8	161.4	183.9	166.4	232.7	191.5

Figure 6.2: Inserting the data in the table.

3. Save the spreadsheet. Once you have entered your data, navigate to the File menu and select the Save As option. This spreadsheet can be saved in various formats, such as ODS, XLSX, and more. If you intend to perform operations with the data, it is essential to save it in one of these formats, as they provide the necessary functionality. However, for **XMapsLab**, it is mandatory to save the spreadsheet in CSV format.

   By default, the program saves the file in the folder where it is run, with the name Untitled 1 and the format as ODS (Figure 6.3). To save the file in a specific folder, navigate to that folder first. Then, click on the Name field to change the name to project_data.csv (or other name, i.e. first_measurement.csv or copy_version.csv, etc.). As you type the .csv extension, the format will automatically change accordingly. Before clicking the Save button, it is important to select the option Edit filter settings to define the separator character. Once all the necessary settings are in place, click the Save button (Figure 6.4).

4. Confirm the CSV format. A window will appear notifying you about the format in which the file will be saved. In this window, click on the option Use text CSV format (Figure 6.5).

5. Set the CSV options. The following window will display options for saving the data. Select Unicode (UTF-8) for the Character set field. The crucial field is Field delimiter, which determines the character used to separate the tabular data. In our case, we want to use the semicolon character. Once all the fields are correctly configured, click the OK button to save the file (Figure 6.6). And that's it!

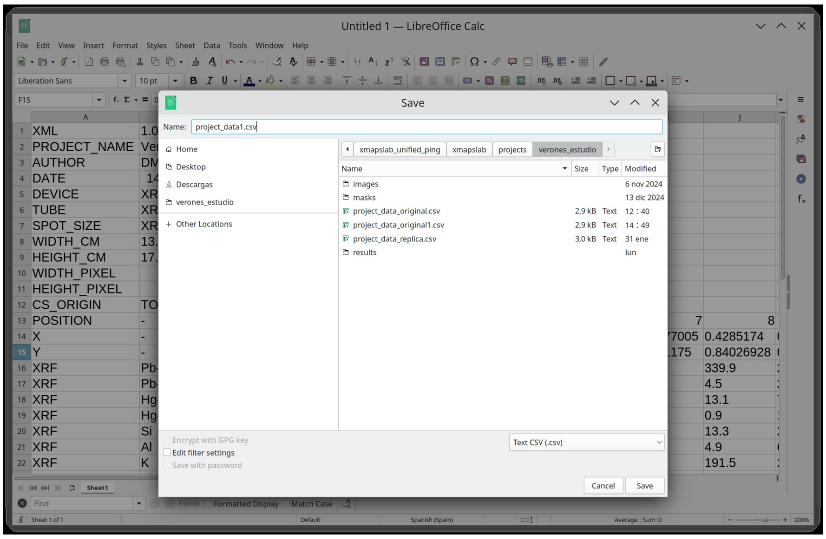


Figure 6.3: Save options.

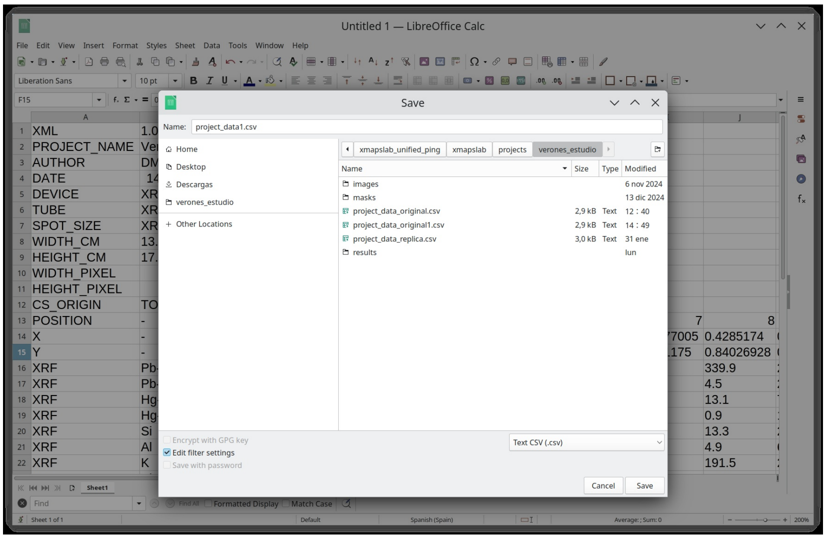


Figure 6.4: The values that must be changed.

Loading a CSV file in the spreadsheet is even easier. One option is to double-click on the file name. LibreOffice will launch, and a window will open displaying information for importing the data. At the bottom of the window, there will be a preview of the importation. The important parameters are found in the Separator Options section. We need to select the Semicolon option and uncheck the other boxes (Figure 6.6). Once we have verified that the preview is correct, we can click the OK button, and the file will be loaded correctly.

In the case of having Excel or another spreadsheet software as the primary spreadsheet program on the system instead of LibreOffice, we will need to follow different steps to load the file. First, we need to launch the LibreOffice program. Then, we will navigate to the

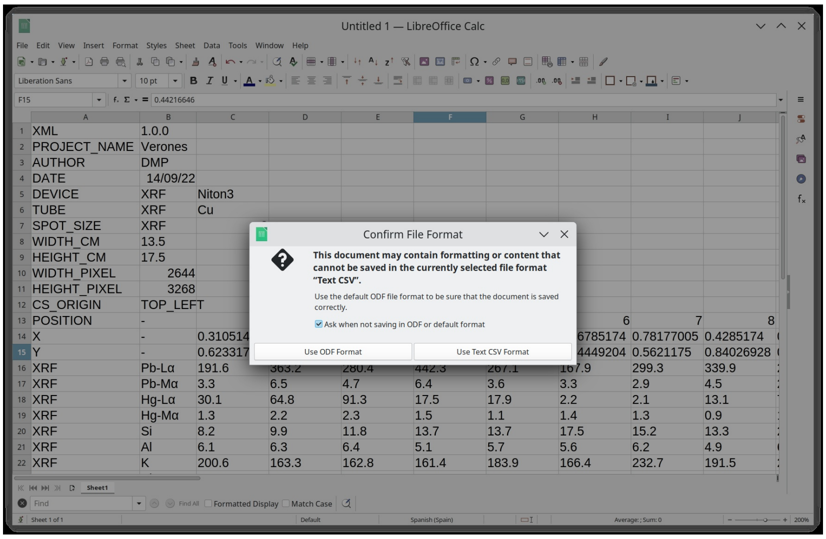


Figure 6.5: Window to confirm the type of the file. It must be CSV.

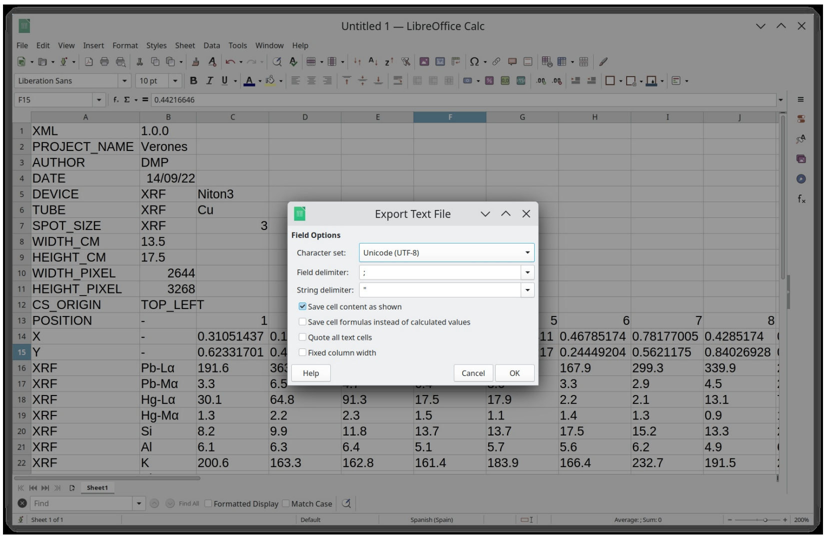


Figure 6.6: Window with the saving options. The most important is the Field delimiter.

Open option, which can be found under the File menu. This will open a window displaying the files in the folder where LibreOffice is located.

We will navigate to the folder where our CSV file is located and click on its name. This action will open the load options window. From there, the process is the same as explained previously.

If you are using Excel, the operations for working with CSV files should be similar. You can open the program and navigate to the Open option, which allows you to select the CSV file from the desired folder. Excel will prompt you with import options, including the choice of delimiter and character encoding. Once you have specified the appropriate settings, you can proceed with opening the file and working with the data in the spreadsheet.

Bibliography

Franz Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, september 1991. doi: 10.1145/116873.116880.

B. Delaunay. Sur la sphère vide. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathematiques et na*, 1934(6):793–800, 1934.

Ronald L Graham and F Frances Yao. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4(4):324–331, 1983. ISSN 0196-6774. doi: https://doi.org/10.1016/0196-6774(83)90013-5. URL https://www.sciencedirect.com/science/article/pii/0196677483900135.

Rolland L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research (1896-1977)*, 76(8):1905–1915, 1971. doi: 10.1029/JB076i008p01905.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Domingo Martín, Germán Arroyo, Juan Ruiz de Miras, Luis López, María Rosario Blanc, Philippe Sarrazin, and Juan Carlos Torres. Evaluation of interpolation methods for generating maps in cultural heritage chemical applications. *Journal of Cultural Heritage*, 39:260–269, 2023. ISSN 1296-2074. doi: 10.1016/j.culher.2019.04.003.

J. Daniel Martín-Ramos and Giacomo Chiari. SmART_scan: A method to produce composition maps using any elemental, molecular and image data. *Journal of Cultural Heritage*, 39:260–269, 2019. ISSN 1296-2074. doi: 10.1016/j.culher.2019.04.003.

A.F. Möbius and Johann Ambrosius Barth. *Der barycentrische Calcul ein neues Hülfsmittel zur analytischen Behandlung der Geometrie dargestellt und insbesondere auf die Bildung neuer Classen von Aufgaben und die Entwickelung mehrerer Eigenschaften der Kegelschnitte angewendet von August Ferdinand Mobius Professor der Astronomie zu Leipzig*. verlag von Johann Ambrosius Barth, 1827. URL https://books.google.es/books?id=ahsoar-z7Z8C.