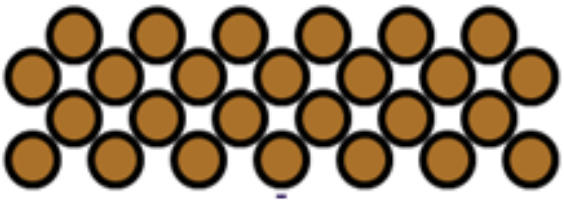
STS TOOLS
Data Analises for Nanosurf
and Omicron



by Rafael Reis Barreto

STStools
Documentation

Version 7.0.0

April 30, 2025

Contents

1. Instalation

1.1 Python, jupyter and Vscode

To use the STStools library, Python must be installed. We recommend **Python version 3.10 or higher**. You can download it from:

Download Python | Python.org

Alternatively, using the Anaconda distribution is **recommended**, as it includes Python and several useful packages.

Download Anaconda Distribution | Anaconda

After installing Python, you will need an environment to run Jupyter Notebooks.

You can use either Jupyter Notebook or Visual Studio Code (VS Code).

⚠ Note: Google Colab is not recommended for this library. To install jupyter notebook go to the link:

Project Jupyter | Installing Jupyter

(VScode is recommendable). To install VS Code:

Download Visual Studio Code - Mac, Linux, Windows

If you're using VS Code, you must also install the **Jupyter extension**.

Follow this tutorial to get started

Jupyter Notebook no VSCode! Como Instalar?

1.2 STStools Library instalation

Installing the STStools library is very simple.

Open the Command Prompt (Windows), Terminal (Linux), or Anaconda Prompt (both), and enter the following command:

```
pip install ststools
```

It will be something like this:

```
C:\Users\rafin>pip install ststools
```

This will start the installation process. Once complete, you should see a confirmation message indicating that the installation was successful.

```
Installing collected packages: ststools
  Running setup.py install for ststools ... done
Successfully installed ststools-6.6.0
```

To check the latest available version of STStools, visit the library's page on PyPI:

ststools · PyPI

To install a specific version (e.g., 7.0.0), use:

```
pip install ststools==7.0.0
```

If you need to reinstall the same version, use:

```
pip install --force-reinstall ststools==7.0.0
```

Note 1: If the installation doesn't work, try uninstalling and reinstalling:

```
pip uninstall ststools
pip install ststools
```

Note 2: To verify the latest version of the library, always refer to the ststools · PyPI

2. STSTOOLS LIBRARY FUNCTIONS

2.1 Import and load files

To begin using the STStools library, you need to import it in a notebook cell by typing:

```
import ststools as sts
```

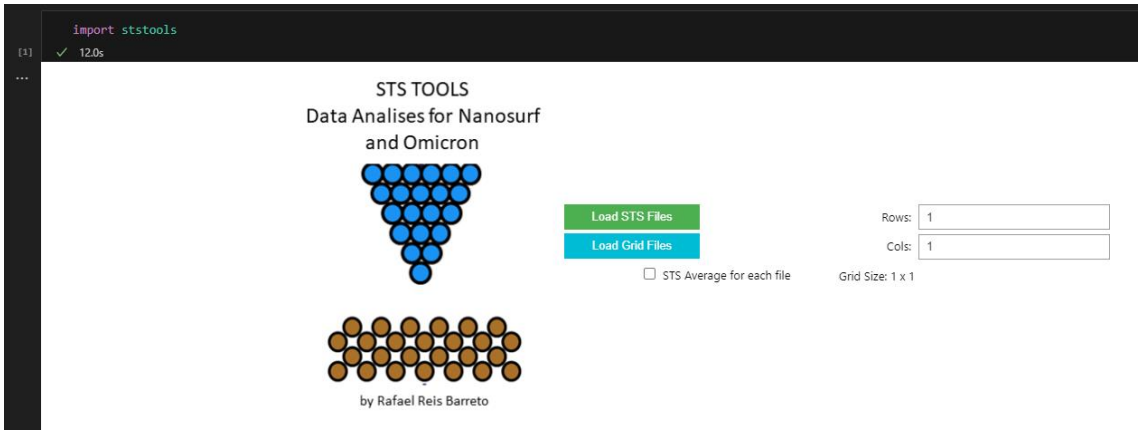
Or simply:

```
import ststools
```

Upon importing the library, a logo and a set of load buttons will appear, along with the following message (translated):

```
Software de visualizacao de arquivos de STS Nanosurf e Omicron v6.6.0
By Rafael Reis Barreto, contato rafinhareis17@gmail.com
Github com o codigo fonte https://github.com/rafinhareis/ststools
Repositorio Pypi https://pypi.org/project/ststools/ (nele voce pode conferir qual a versao mais atual)
Arquivos Nanosurf tipo .nid
Aquivos Omicron tipo .txt, ou .ibw (IGOR) para grid. NAO MUDAR O NOME DO ARQUIVO PADRAO
Arquivos Nanonis tipo .dat
Se sentir no fundo do coracaozinho, poe meu nome no artigo =D
```

When the interface is loaded, it will look like this:



2.1.1 Load STS Files Button

This button is used exclusively for loading STS (scanning tunneling spectroscopy) data.

Although you may open grid files with this button, only the **Display** function will be available for individual STS curve analysis.

⚠**OBS**: To access the Map function for spatial data visualization, you must use the "Load Grid Files" button instead.

Once the files are successfully loaded, a confirmation message — “STS files loaded” — will appear.

2.1.2 Load Grid Files Button

This button is used to open grid files or STS files that collectively form a grid. It functions similarly to the Load STS Files button but offers additional features, including spatial mapping.

You can still use the **Display** function to analyze STS curves loaded via this button.

Note: The reason for having two separate buttons is that grid data requires a different loading strategy, involving more computational resources and time — especially when dealing with a large number of files.

Therefore, if your goal is to analyze only the STS data (even if the original file comes from a grid), it's recommended to use the STS File button for faster performance.

Once loading is complete, a confirmation message — “Grid files loaded” — will appear.

.

2.1.3 STS Average for Each File

This checkbox enables the option to **average the STS curves within each individual file** before loading.

For example, suppose you have **10 files**, and each file contains **5 STS curves**.

- If the checkbox **is not selected** (default), all curves are loaded individually, resulting in **10 × 5 = 50 STS curves**.
- If the checkbox **is selected**, the software will compute the **average curve for each file**, resulting in only **10 STS curves**, each representing the average of the 5 curves in its respective file.

☑ This feature is useful for simplifying analysis by reducing curve count and emphasizing global trends within each file

2.1.4 Rows and Columns

This option is only applicable when working with **GRID files**. If you intend to build a grid using **multiple individual files**, you must manually define the grid dimensions by entering the values in the **Rows** and **Cols** input boxes **before** loading the files.

Example:
If you want to create a 5×5 grid using 25 files, you must first set **Rows = 5** and **Cols = 5**, and then click the **Load Grid Files** button.

It is essential that **each of the 25 files contains only one STS curve**. If they contain more than one, enable the **"STS Average for Each File"** option to reduce them to one averaged curve per file.

Note:
This method of building grids from multiple files applies to **all supported formats**: **OMICRON**, **NANOSURF**, and **NANONIS**.

If you are using a **single grid file**, behavior differs slightly based on the format:

- **OMICRON**: The file must be in **IGOR (.ibw)** format. In this case, set **Rows = 1** and **Cols = 1**.
- **NANOSURF**: If using a single **.nid** file containing a grid (e.g., 5×5), you **must** manually enter **Rows = 5** and **Cols = 5** before loading.

In the next subsection, the supported file types will be discussed in more detail to clarify compatibility and behavior.

2.1.5 Supported Files Types

The **STStools** library supports data files from three different scanning probe microscopy systems:

- **OMICRON**
- **NANOSURF**
- **NANONIS**

Below are the supported file formats for each system:

STS Files (single-point spectroscopy):

Microscope	File Format
OMICRON	.txt
NANOSURF	.nid
NANONIS	.dat

GRID Files (spatially resolved maps):

Microscope	File Format
OMICRON	.ibw (single file) .txt (multiple files)
NANOSURF	.nid (single or multiple files)
NANONIS	.dat (multiple files)

Important Notes (based on section 2.1.4):

STStools allows grids to be opened in **two ways**:

- **Single file** containing the entire grid
- **Multiple individual files** forming a grid

🔹 *For Single Grid Files:*

- **OMICRON (.ibw)**:
  - Must be exported from **VERNISAGE** in IGOR format.
  - Grid size and STS curves are embedded in the file.
  - You do **not** need to set Rows and Cols — just load the file using the **Load Grid Files** button, with Rows and Cols set to **1×1**.
- **NANOSURF (.nid)**:
  - This is the default file format.
  - However, **grid size is not embedded**.
  - You **must manually enter** the number of rows and columns (e.g., 5×5) **before loading** the file using the **Load Grid Files** button.

🔹 *For Multiple Files Forming a Grid:*

- This approach works **the same for all supported microscopes**.
- You must:
  1. Set **Rows and Cols** manually (e.g., 10×10 for 100 files).
  2. Use the **Load Grid Files** button to select and open all files.
  3. If each file contains **more than one STS curve**, you **must enable** the **"STS Average for Each File"** checkbox to reduce them to a single averaged curve per file.

2.2 Display Function - Data Visualization

The **Display()** function is one of the core components of the STStools library. It is used to **visualize STS data interactively** and provides tools to **analyze, inspect, and export** data for further use in external software such as **Origin**.

To activate the display interface, simply type in a notebook cell:

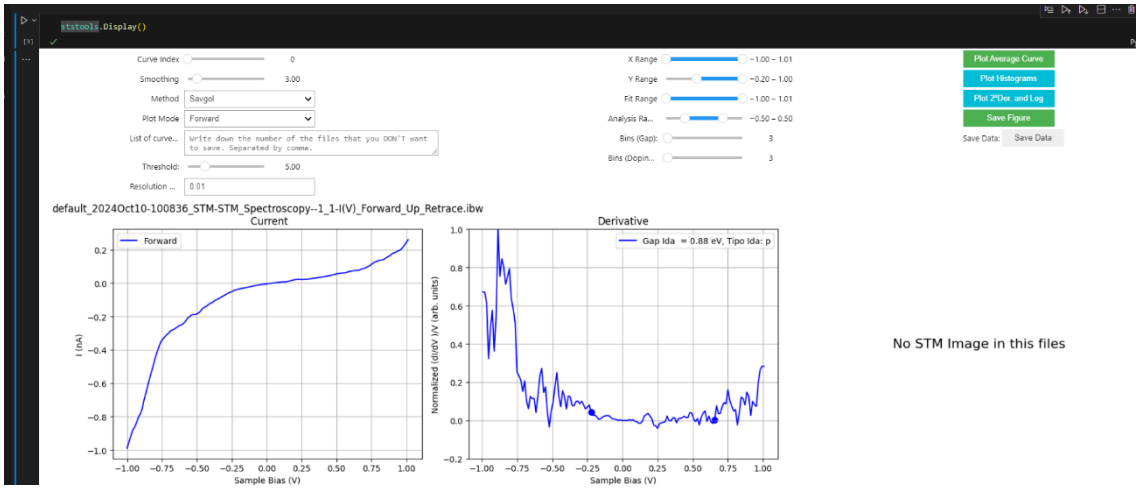
```
sts.Display()
```

Or

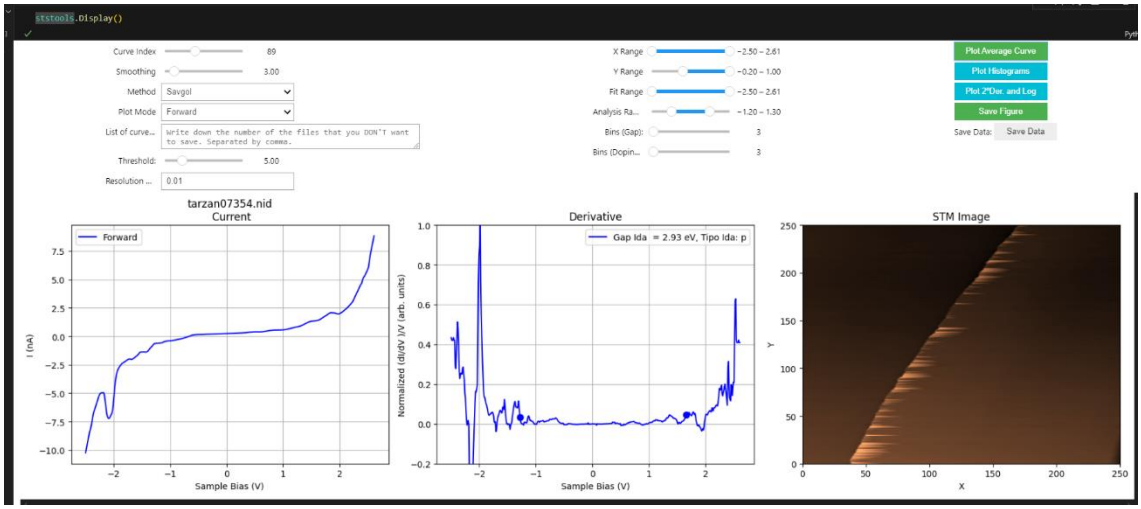
```
ststools.Display()
```

The Interface

Once launched, the interface will appear as shown below:



If an STM image is available (as in the case of NANOSURF files), it will be displayed on the right side of the interface.



At the **top** of the interface, you will find a set of control tools and options. Each of these will be described in detail in the following subsections.

Main Visualization Panels:

- **Left Panel:** Shows the **I×V (current vs. voltage)** curve.
- **Right Panel:** Shows the **dI/dV (differential conductance)** curve.

Additional Features:

- The **file name** containing the plotted STS curve is shown above the I×V plot.
- The **dI/dV plot** includes information about the **energy gap** and **doping level**, displayed as labels inside the graph.
- For **NANOSURF** data, a corresponding **STM image** will also appear, offering spatial context.

ℹ **Note:** A marker (dot) is shown over the dI/dV curve to indicate the point where the **gap is calculated**.

2.2.1 Curve Index:

This slider allows you to **select which STS curve to visualize** from the loaded dataset.

When you change the curve index, the corresponding curve will be updated in both the **I×V** and **dI/dV** plots.

- The **file name** containing the selected curve is displayed above the I×V plot.
- If a file contains **multiple curves**, the file name will remain the same for each, but the curves will be distinguished by their **index number**.

ℹ **Note:** This is especially relevant when multiple STS curves are stored in a single file (e.g., multi-curve .nid, .txt, or .dat files).

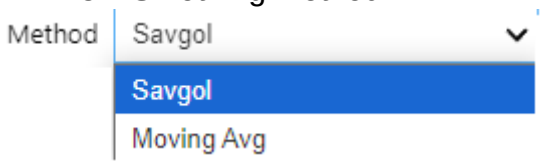
2.2.2 Smoothing

This slider controls the **amount of smoothing applied to the I×V curve**, expressed as a **percentage** of the curve length. The default value is **3%**.

Smoothing can help reduce noise and make features like peaks or gaps more visually clear.

⚠ **Note:** Use caution when applying values **greater than 10%**, as excessive smoothing may **distort or remove meaningful features** from the STS curve, potentially leading to misinterpretation of the data.

2.2.3 Smothing Method



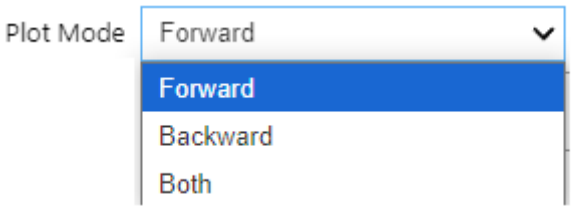
This dropdown menu allows you to select the **smoothing algorithm** applied to the STS curve.

There are two available methods:

- **Savgol** — *(Savitzky-Golay filter)*: A polynomial-based smoothing technique that preserves curve features such as peaks and inflection points. **Recommended** for most spectroscopy data. *(Default option)*
- **Moving Average:** A simple averaging method that smooths fluctuations by averaging over neighboring points. Less sensitive to outliers but may flatten sharp features.

ℹ **Tip:** Start with **Savgol** for best results and only switch to Moving Average if needed for your specific dataset.

2.2.4 Plot Mode

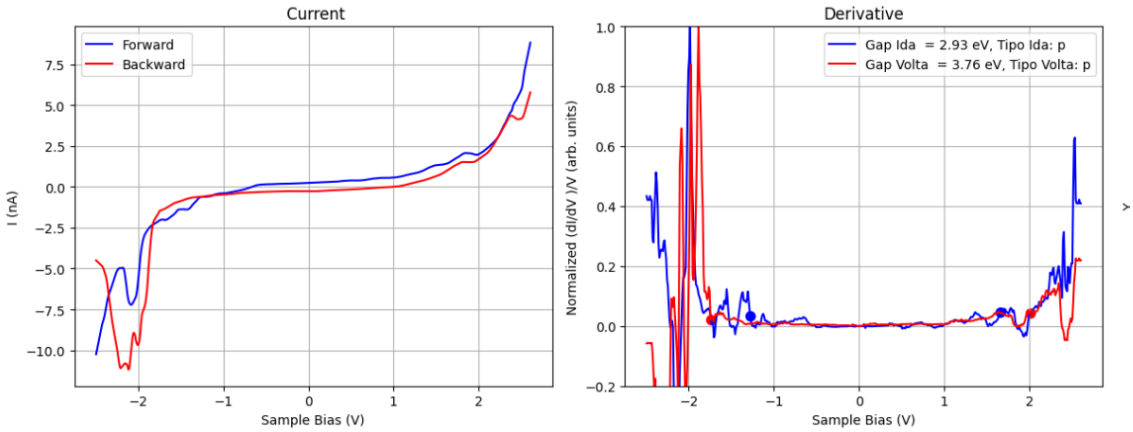


This dropdown menu lets you choose which **acquisition direction** of the STS curve to visualize. The available options are:

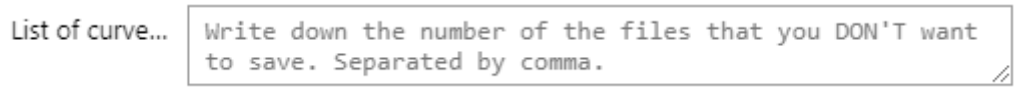
- **Forward** *(default)* – Shows the curve recorded during the forward voltage sweep.
- **Backward** – Shows the curve recorded during the reverse voltage sweep.
- **Both** – Displays both forward and backward curves simultaneously for comparison.

ℹ **Note:** If the data file contains **only forward curves**, the backward curve will be a **copy of the forward**.

When **Both** is selected, the two curves will be displayed together for visual comparison, helping to identify hysteresis or measurement asymmetries.



2.2.5 List of Curves to Be Saved



This input box allows you to **exclude specific STS curves** from being saved. You should enter the **curve indices you do *not* want to export**, separated by commas.

🔷 **Example:** If you have 10 curves, but curves with indices **2** and **4** are noisy or invalid, simply enter:

When you proceed to save the data, only the remaining curves (**0, 1, 3, 5, 6, 7, 8, 9**) will be included in the exported files.

ℹ **Note:** This exclusion only affects the export operation; the full dataset remains loaded for inspection.

2.2.6 Threshold

This slider sets the **threshold value** used to calculate the **energy gap** and **doping level**, expressed as a **percentage of the maximum value** of the **dI/dV curve**.

🔷 **Example:** If the threshold is set to **5%**, then any portion of the dI/dV curve that falls **below 5% of its maximum value** will be considered part of the **energy gap**.

This method allows for a consistent and objective gap estimation across curves, particularly useful when analyzing large datasets.

⚠️ **Note:** Adjusting the threshold can significantly affect the calculated gap and doping values. Use lower percentages for high-resolution data, and increase only if noise is masking relevant features.

2.2.7 Resolution

This input box defines the **voltage resolution** used to determine whether a doping value is considered **neutral**.

🔷 **Example:**
If the resolution is set to **0.01 V** (default), then any difference between the negative and positive bias voltages (used in the doping calculation) that is **less than 0.01 V** will be classified as **neutral doping**.

This parameter helps to avoid misclassifying very small voltage shifts as meaningful doping. It is particularly useful when working with symmetric or nearly symmetric dI/dV curves.

ℹ️ **Note:** You can increase this value for noisy data or decrease it for high-precision measurements.

2.2.8 Scale X and Y Axis

These two sliders allow you to manually adjust the **x-axis** and **y-axis** scales of the **dI/dV curve** plot.

- The **X-axis scale** controls the voltage range displayed.
- The **Y-axis scale** adjusts the vertical range of the differential conductance values.

This feature is useful for focusing on specific regions of interest or enhancing visibility when analyzing small signal variations.

ℹ **Tip:** Adjusting the axis scales can help reveal subtle features that might otherwise be hidden in a full-range plot.

2.2.9 Fit and Analysis

These two sliders define the ranges used to **fit a hyperbolic tangent function** to the **second derivative curve** (*d^2I/dV).*

- **Fit Range:** Specifies the **full voltage window** (x-axis) over which the fit will be applied.

  Example: -1 V to 1.01 V

- **Analysis Range:** Defines the **regions within the fit range** that will actually be used for the fitting procedure. This range excludes points near zero, focusing only on the outer parts of the curve.

  Example: The fit will be applied from -1 V to -0.5 V (negative side) and from 0.5 V to 1.01 V (positive side).

This selective fitting avoids noisy regions near zero bias and improves the reliability of extracted physical parameters such as the gap.

ℹ **Tip:** Adjust the analysis range to exclude flat or noisy central regions while preserving edge information for accurate fitting.

2.2.10 Bins for Gap and Dopping Histogram

These two sliders control the **number of bins** used to generate the **histograms** for gap and doping distributions.

By default, the number of bins is determined automatically using a combination of two standard statistical methods:

- **Sturges' Rule**: Based on the logarithm of the number of data points. Ideal for **smaller datasets** (*typically < 100*) and **normally distributed data**.

- **Freedman–Diaconis Rule**: Calculates bin width based on the **interquartile range (IQR)** and is better suited for **larger datasets** or **data with outliers and skew**.

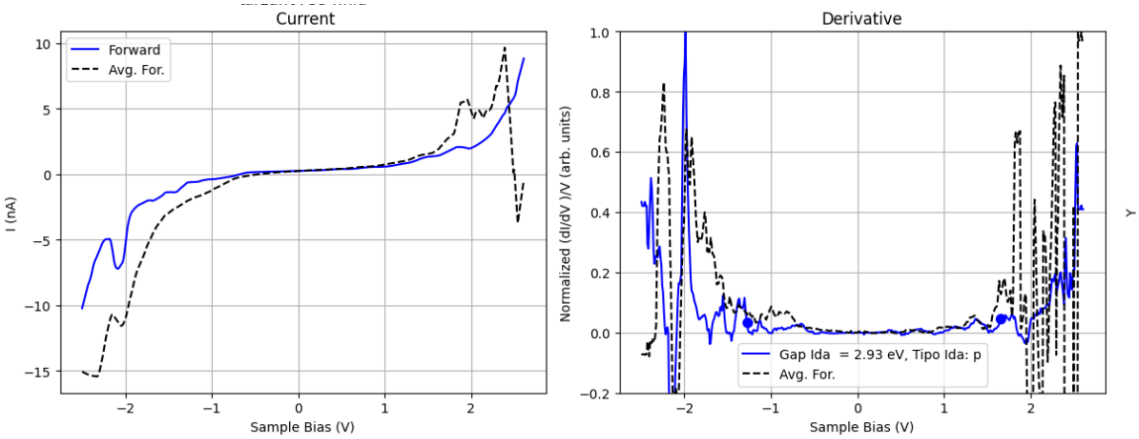
You can override the default settings using the sliders to manually define the number of bins for each histogram, giving you more control over the granularity of the analysis.

ℹ **Tip:** Use more bins to reveal finer structure in your data; use fewer bins to highlight general trends.

2.2.11 Plot Average Curve

Clicking this button will **calculate and overlay the average STS curve** across all loaded data. For example, if you have 10 curves loaded, this feature will compute and display the **average of those 10 curves**, plotted together with the currently selected curve index on all plots.

This feature works across **all plot modes** (Forward, Backward, Both), and helps identify overall trends or behaviors in your dataset.



Behavior:

- Once activated, the button's **green color darkens** to indicate that the average curve is being displayed.
- Clicking the button again will **deactivate the average overlay**, returning the plots to their original state.
- The average is computed **at the moment the button is pressed** — so if you have a large number of curves, it may take a few seconds to process.

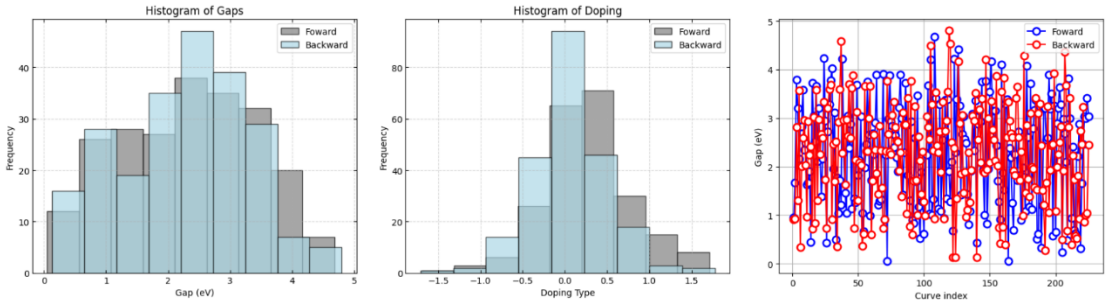
💡 **Tip:** Only press this button when you wish to **temporarily view** the average. If you no longer need to see it, **leave the button unpressed** (lighter green) to maintain performance.

2.2.12 Plot Histogram

Click this button to **calculate and display histograms** of the **gap** and **doping values** for all loaded STS curves.

Upon activation, the tool will:

- Compute the **gap and doping** for both **forward** and **backward** curves (if available).
- Display a **histogram** showing the distribution of gap and doping values.
- Generate an additional plot showing the **relationship between curve index** and the **calculated gap**, providing insight into spatial or sequential variation in the dataset.



Behavior:

- When the button is activated, its **blue color darkens** to indicate that the histogram is being displayed.
- Click the button again to **hide** the histogram and return to the default view.
- Gap and doping values are calculated **in real time when the button is pressed**, so the computation **may take a few seconds** if many curves are loaded.

💡 **Tip:** Press the button only when you wish to visualize the histogram. To disable it and improve responsiveness, leave the button **unpressed** (lighter blue).

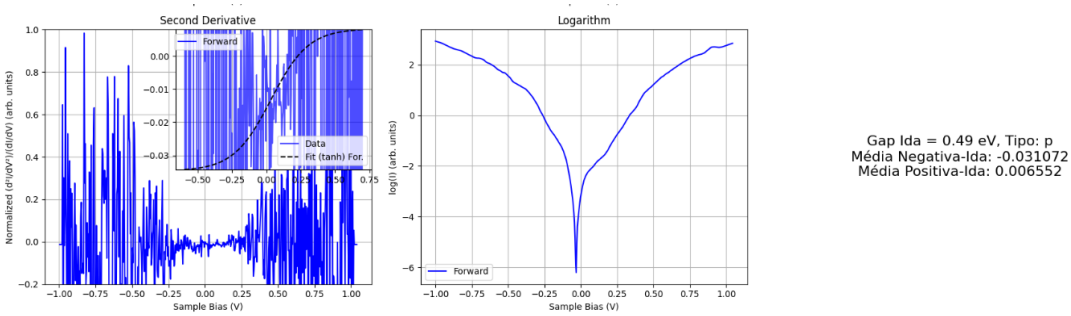
2.2.13 Plot 2º Derivative and Log

Plot 2ºDer. and Log

Click this button to **calculate and display** the **second derivative** (d^2I/dV^2) and the **logarithmic plot** (*log(I) vs V*) of the selected STS curve.

Upon activation, the tool performs the following:

- Computes the **second derivative** of the I×V curve.
- Attempts to **fit a hyperbolic tangent function** to the second derivative curve.
- Displays:
  - **Left panel**: The second derivative plot (d^2I/dV^2), with an inset showing the fit curve and the region used for fitting (if successful).
  - **Middle panel**: The log plot of the STS data (*log(I) vs V*).
  - **Right panel**: A summary containing the **calculated gap**, **doping value**, and the **average values of the second derivative** for both the negative and positive voltage regions.



Behavior:

- When the button is pressed, it turns **dark blue**, indicating that the derivative and log views are active.
- Click the button again to **disable** these plots and return to the standard view.
- The fitting and plotting are calculated **on demand**, so it may take a few seconds for large or noisy datasets.

💡 **Tip:** Use this tool for in-depth analysis, but keep the button **unpressed** (lighter blue) when not needed to improve performance and clarity.

2.2.14 Save Figure

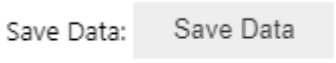
Save Figure

Click this button to **save the current figure** as a .png image file.

- The image will include the **currently displayed plots**, such as the I×V curve, dI/dV curve, and any overlays (e.g., average, fit).
- The **filename of the STS curve** currently being visualized (shown above the I×V plot) will be used as the **name of the saved image**.
- The .png file will be saved **in the same directory** where the Jupyter Notebook is running.

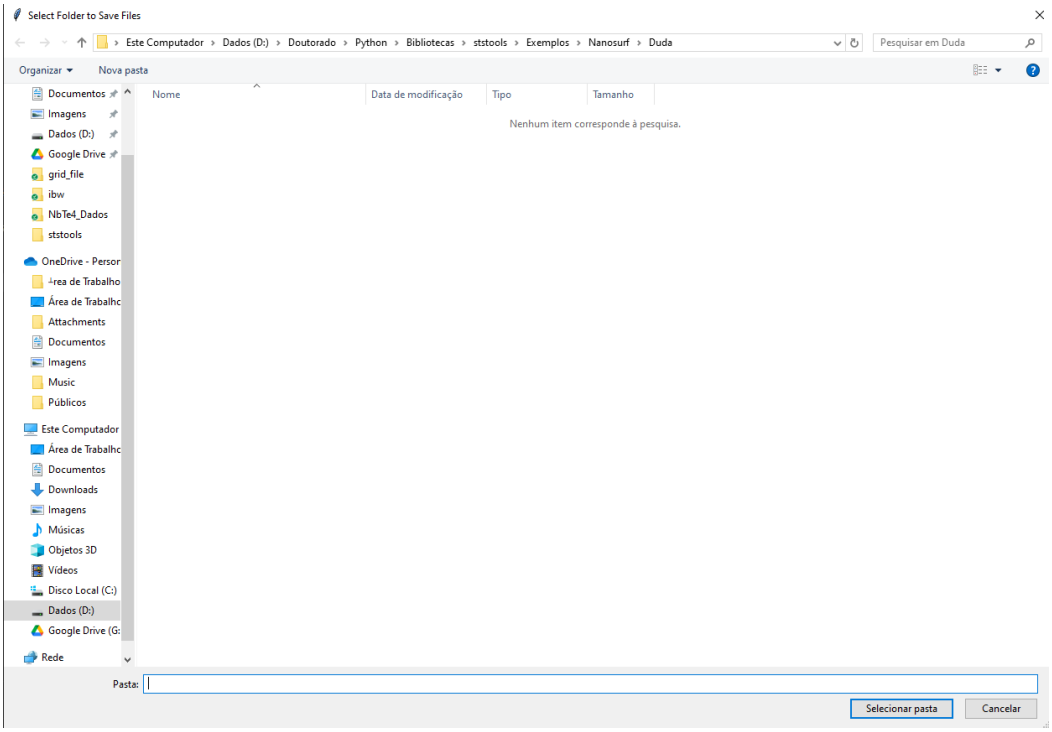
💡 **Tip:** Use this feature to quickly document specific curves or analysis results for inclusion in reports or publications.

2.2.15 Save Data



Click this button to **export all analysis results**. When pressed, a file dialog window will appear, allowing you to select the **folder (directory)** where the files should be saved.

⚠️ **Important:** On some systems, the file dialog may open in the **background**. If you don't see it immediately, use **Alt+Tab** or check your taskbar to bring it to the front.

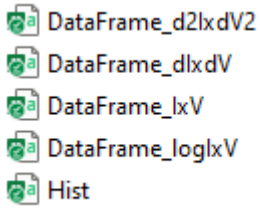


📂 Files Saved

The following .csv files will be generated in the selected folder:

- **Dataframe_IxV.csv** — Contains all I×V curves (columns: voltage and current). Excludes any curve indices specified in the *"List of curves to be saved"* box.
- **Dataframe_dIxdV.csv** — Contains all dI/dV curves. Same exclusion rules apply.
- **Dataframe_d2IxdV2.csv** — Contains all second derivative curves (d^2I/dV^2). Same exclusion rules apply.
- **Dataframe_logIxV.csv** — Contains all log(I) vs V curves. Same exclusion rules apply.
- **Hist.csv** — Contains histogram data with:
    - Column 1: Gap values
    - Column 2: Doping values
    - Column 3: Doping type classification (e.g., n-type, p-type, neutral)

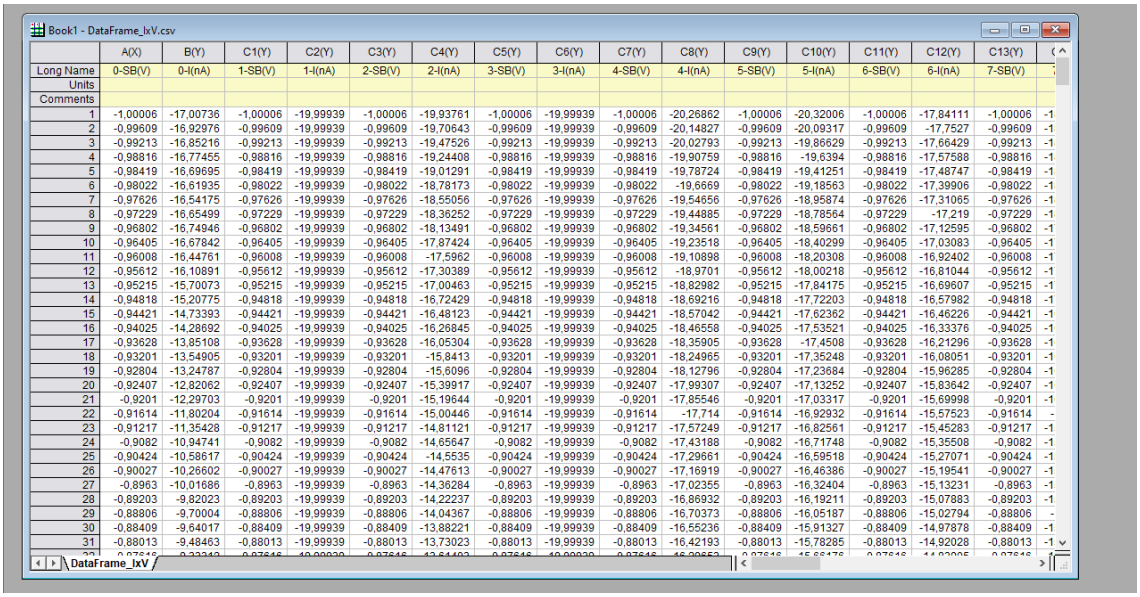
It should appear like this:



Each file includes a **header section** with metadata:

- % STS tools Version: — the software version used to export the data.
- % Map type: — the type of measurement (e.g., *Current Map (nA)*).
- % Export Date: — date and time the data was exported.

After loaded to origin, it should appear like this:



Book1 - DataFrame_IxV.csv

	A(X)	B(Y)	C1(Y)	C2(Y)	C3(Y)	C4(Y)	C5(Y)	C6(Y)	C7(Y)	C8(Y)	C9(Y)	C10(Y)	C11(Y)	C12(Y)	C13(Y)
Long Name	0-SB(V)	0-I(nA)	1-SB(V)	1-I(nA)	2-SB(V)	2-I(nA)	3-SB(V)	3-I(nA)	4-SB(V)	4-I(nA)	5-SB(V)	5-I(nA)	6-SB(V)	6-I(nA)	7-SB(V)
Units															
Comments															
1	-1,00006	-17,00736	-1,00006	-19,99939	-1,00006	-19,93761	-1,00006	-19,99939	-1,00006	-20,26862	-1,00006	-20,32006	-1,00006	-17,84111	-1,00006
2	-0,99609	-16,92976	-0,99609	-19,99939	-0,99609	-19,70643	-0,99609	-19,99939	-0,99609	-20,14827	-0,99609	-20,09317	-0,99609	-17,7527	-0,99609
3	-0,99213	-16,85216	-0,99213	-19,99939	-0,99213	-19,47526	-0,99213	-19,99939	-0,99213	-20,02793	-0,99213	-19,86629	-0,99213	-17,66429	-0,99213
4	-0,98816	-16,77455	-0,98816	-19,99939	-0,98816	-19,24408	-0,98816	-19,99939	-0,98816	-19,90759	-0,98816	-19,6394	-0,98816	-17,57588	-0,98816
5	-0,98419	-16,69695	-0,98419	-19,99939	-0,98419	-19,01291	-0,98419	-19,99939	-0,98419	-19,78724	-0,98419	-19,41251	-0,98419	-17,48747	-0,98419
6	-0,98022	-16,61935	-0,98022	-19,99939	-0,98022	-18,78173	-0,98022	-19,99939	-0,98022	-19,6669	-0,98022	-19,18563	-0,98022	-17,39906	-0,98022
7	-0,97626	-16,54175	-0,97626	-19,99939	-0,97626	-18,55056	-0,97626	-19,99939	-0,97626	-19,54656	-0,97626	-18,95874	-0,97626	-17,31065	-0,97626
8	-0,97229	-16,65499	-0,97229	-19,99939	-0,97229	-18,36252	-0,97229	-19,99939	-0,97229	-19,44885	-0,97229	-18,78564	-0,97229	-17,219	-0,97229
9	-0,96802	-16,74946	-0,96802	-19,99939	-0,96802	-18,13491	-0,96802	-19,99939	-0,96802	-19,34561	-0,96802	-18,59661	-0,96802	-17,12595	-0,96802
10	-0,96405	-16,67842	-0,96405	-19,99939	-0,96405	-17,87424	-0,96405	-19,99939	-0,96405	-19,23518	-0,96405	-18,40299	-0,96405	-17,03083	-0,96405
11	-0,96008	-16,44761	-0,96008	-19,99939	-0,96008	-17,5962	-0,96008	-19,99939	-0,96008	-19,10898	-0,96008	-18,20308	-0,96008	-16,92402	-0,96008
12	-0,95612	-16,10891	-0,95612	-19,99939	-0,95612	-17,30389	-0,95612	-19,99939	-0,95612	-18,9701	-0,95612	-18,00218	-0,95612	-16,81044	-0,95612
13	-0,95215	-15,70073	-0,95215	-19,99939	-0,95215	-17,00463	-0,95215	-19,99939	-0,95215	-18,82982	-0,95215	-17,84175	-0,95215	-16,69607	-0,95215
14	-0,94818	-15,20775	-0,94818	-19,99939	-0,94818	-16,72429	-0,94818	-19,99939	-0,94818	-18,69216	-0,94818	-17,72203	-0,94818	-16,57982	-0,94818
15	-0,94421	-14,73393	-0,94421	-19,99939	-0,94421	-16,48123	-0,94421	-19,99939	-0,94421	-18,57042	-0,94421	-17,62362	-0,94421	-16,46226	-0,94421
16	-0,94025	-14,28692	-0,94025	-19,99939	-0,94025	-16,26845	-0,94025	-19,99939	-0,94025	-18,46558	-0,94025	-17,53521	-0,94025	-16,33376	-0,94025
17	-0,93628	-13,85108	-0,93628	-19,99939	-0,93628	-16,05304	-0,93628	-19,99939	-0,93628	-18,35905	-0,93628	-17,4508	-0,93628	-16,21296	-0,93628
18	-0,93201	-13,54905	-0,93201	-19,99939	-0,93201	-15,8413	-0,93201	-19,99939	-0,93201	-18,24965	-0,93201	-17,35248	-0,93201	-16,08051	-0,93201
19	-0,92804	-13,24787	-0,92804	-19,99939	-0,92804	-15,6096	-0,92804	-19,99939	-0,92804	-18,12796	-0,92804	-17,23684	-0,92804	-15,96285	-0,92804
20	-0,92407	-12,82062	-0,92407	-19,99939	-0,92407	-15,39917	-0,92407	-19,99939	-0,92407	-17,99307	-0,92407	-17,13252	-0,92407	-15,83642	-0,92407
21	-0,9201	-12,29703	-0,9201	-19,99939	-0,9201	-15,19644	-0,9201	-19,99939	-0,9201	-17,85546	-0,9201	-17,03317	-0,9201	-15,69998	-0,9201
22	-0,91614	-11,80204	-0,91614	-19,99939	-0,91614	-15,00446	-0,91614	-19,99939	-0,91614	-17,714	-0,91614	-16,92932	-0,91614	-15,57523	-0,91614
23	-0,91217	-11,35428	-0,91217	-19,99939	-0,91217	-14,81121	-0,91217	-19,99939	-0,91217	-17,57249	-0,91217	-16,82561	-0,91217	-15,45283	-0,91217
24	-0,9082	-10,94741	-0,9082	-19,99939	-0,9082	-14,65647	-0,9082	-19,99939	-0,9082	-17,43188	-0,9082	-16,71748	-0,9082	-15,35508	-0,9082
25	-0,90424	-10,58617	-0,90424	-19,99939	-0,90424	-14,5535	-0,90424	-19,99939	-0,90424	-17,29661	-0,90424	-16,59518	-0,90424	-15,27071	-0,90424
26	-0,90027	-10,26602	-0,90027	-19,99939	-0,90027	-14,47613	-0,90027	-19,99939	-0,90027	-17,16919	-0,90027	-16,46386	-0,90027	-15,19541	-0,90027
27	-0,8963	-10,01686	-0,8963	-19,99939	-0,8963	-14,36284	-0,8963	-19,99939	-0,8963	-17,02355	-0,8963	-16,32404	-0,8963	-15,13231	-0,8963
28	-0,89203	-9,82023	-0,89203	-19,99939	-0,89203	-14,22237	-0,89203	-19,99939	-0,89203	-16,86932	-0,89203	-16,19211	-0,89203	-15,07883	-0,89203
29	-0,88806	-9,70004	-0,88806	-19,99939	-0,88806	-14,04367	-0,88806	-19,99939	-0,88806	-16,70373	-0,88806	-16,05187	-0,88806	-15,02794	-0,88806
30	-0,88409	-9,64017	-0,88409	-19,99939	-0,88409	-13,88221	-0,88409	-19,99939	-0,88409	-16,55236	-0,88409	-15,91327	-0,88409	-14,97878	-0,88409
31	-0,88013	-9,48463	-0,88013	-19,99939	-0,88013	-13,73023	-0,88013	-19,99939	-0,88013	-16,42193	-0,88013	-15,78285	-0,88013	-14,92028	-0,88013

DataFrame_IxV

💡 Notes:

- **SB** refers to *Sample Bias*. Curve names like 0-, 1-, etc., indicate the index of the STS curve the data belongs to.
- To import .csv files into **Origin**, it is recommended to use the **Import Wizard** to prevent incorrect parsing or formatting errors.
- To visualize the **histogram** in Origin, go to the **Statistics > Histogram Plot** section.

2.3 Map function - Map Visualization

The **Map()** function in STStools provides an interactive interface for **visualizing and analyzing spatially resolved STS maps**. This includes maps of current, dI/dV, energy gap, and doping.

It also allows users to **export maps and STS curves** for use in external software such as **Origin**.

To launch the map interface, enter in a notebook cell:

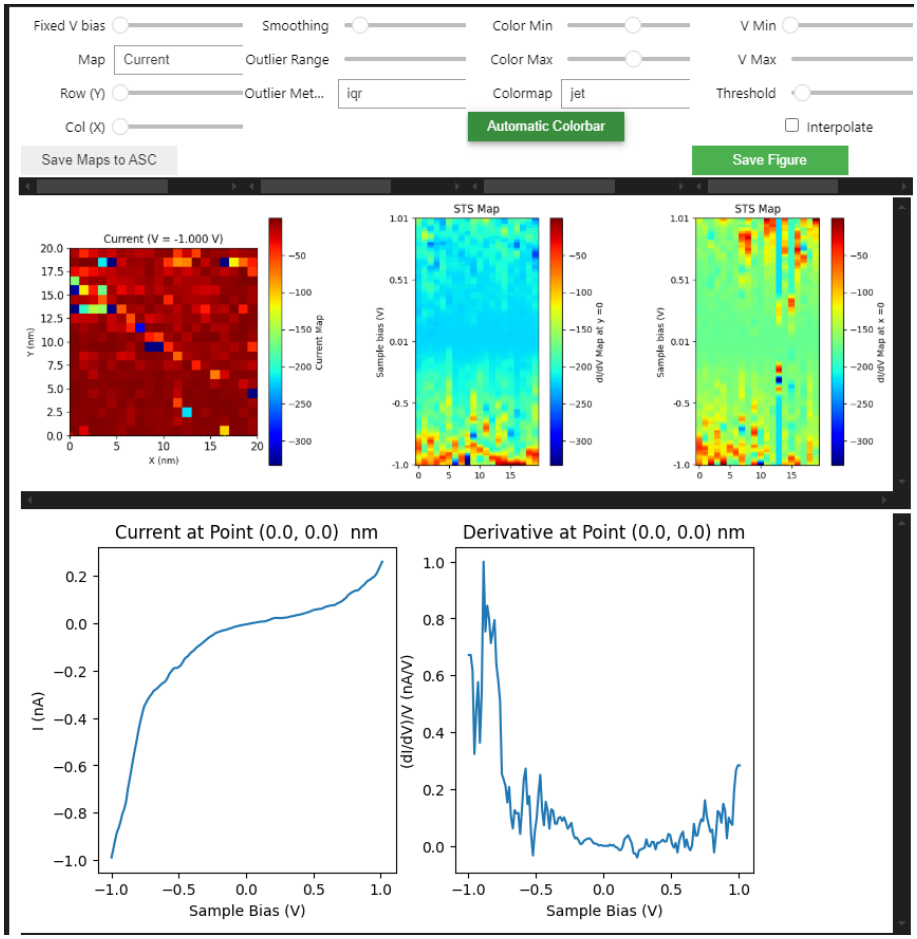
```
sts.Map()
```

Or

```
ststools.Map()
```

Interface Overview

When launched, the interface will appear as shown below:



Layout and Components

At the **top**, a toolbar provides controls for selecting map type, position, colormap, smoothing, and export options. Each control will be explained in detail in the following subsections.

The **main visualization area** consists of several panels:

- **Far Left (if available)** – Displays the **STM image** (for Nanosurf .nid files only).
- **Left Panel** – Shows a **color map** of the selected map type (e.g., *Current Map*, *Gap Map*, etc.). The default is the **Current Map**.
- **Center Panel** – Displays a **dI/dV map along the x-axis**, extracted at a fixed y-position.
- **Right Panel** – Displays a **dI/dV map along the y-axis**, extracted at a fixed x-position.

An **interactive red dot and dashed black lines** indicate the currently selected (x, y) position.

At the **bottom**, two plots are shown:

- **Left**: The I×V curve at the selected position.
- **Right**: The dI/dV curve at the same position.

For **NANOSURF** files, the STM image is automatically included in the visualization.

💡 This interface is particularly useful for correlating spatial information with spectral properties, allowing the user to explore energy gap variations, doping behavior, and other physical properties across the scanned area.

2.3.1 Fixed Bias:

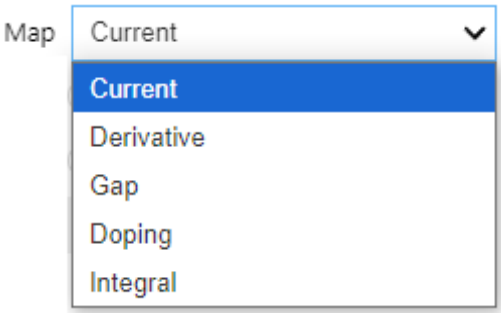
This slider allows you to **select the sample bias voltage (in volts)** at which the **Current Map** and **dI/dV maps** will be displayed.

By adjusting the fixed bias value, you can visualize how the current or differential conductance varies spatially across the sample **at a specific voltage**.

ℹ️ **Unit:** The selected value is in **volts (V)** and corresponds to the horizontal axis of the STS curves.

This tool is especially useful for energy-resolved mapping, where physical properties such as **local density of states**, **gap features**, or **resonances** are voltage-dependent.

2.3.2 Map:



This dropdown menu allows you to **select the type of map** you want to visualize. The available options are:

- **Current** — Maps the tunneling current at the selected fixed bias.
- **Derivative** — Maps the differential conductance (dI/dV) at the selected fixed bias.
- **Gap** — Displays the spatial distribution of the energy gap calculated from the STS curves.
- **Doping** — Shows the spatial distribution of doping values and types.
- **Integral** — Displays the integral of the STS signal over a defined voltage range.

⚠ **Note:** The **Current** and **Derivative** maps depend on the voltage selected via the **Fixed Bias** slider. Other maps (Gap, Doping, Integral) are calculated from the full I×V or dI/dV curve and are not voltage-dependent.

2.3.3 Row and Column:

These input fields allow you to **select the (Row, Column) position** on the grid that you want to visualize.

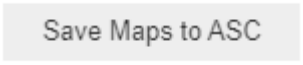
- The selected **row and column** are represented by **dashed black lines** on the map.
- The selected **point (X, Y)** is highlighted with a **red dot**.
- The **I×V** and **dI/dV** curves displayed at the bottom of the interface correspond to this specific spatial point.

Additionally:

- The **dI/dV map across a row** is generated using the selected **column**.
- The **dI/dV map across a column** is generated using the selected **row**.

🛈 This feature allows precise spatial selection and inspection of local STS behavior.

2.3.4 Save Maps to ASC



Click this button to **export all available maps to ASCII (.asc) files**.

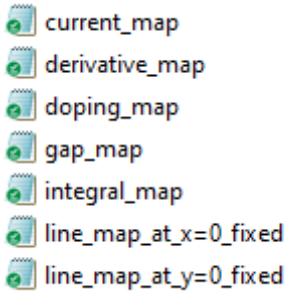
When pressed, the files will be saved in the **same directory** where the *.ipynb* notebook is running.

📂 Files Saved

The following .asc files may be generated depending on the available data:

- **current_map.asc** — Current map at the selected sample bias.
- **derivative_map.asc** — dI/dV map at the selected sample bias.
- **gap_map.asc** — Energy gap map based on the current smoothing and threshold settings.
- **dopping_map.asc** — Doping map, calculated using the current smoothing and threshold values.
- **integral_map.asc** — Integral map computed over the selected voltage range (V_{min} to V_{max}).
- **line_map_at_x=..._fixed.asc** — dI/dV vs. sample bias along the Y-axis at a fixed X (selected column).
- **line_map_at_y=..._fixed.asc** — dI/dV vs. sample bias along the X-axis at a fixed Y (selected row).

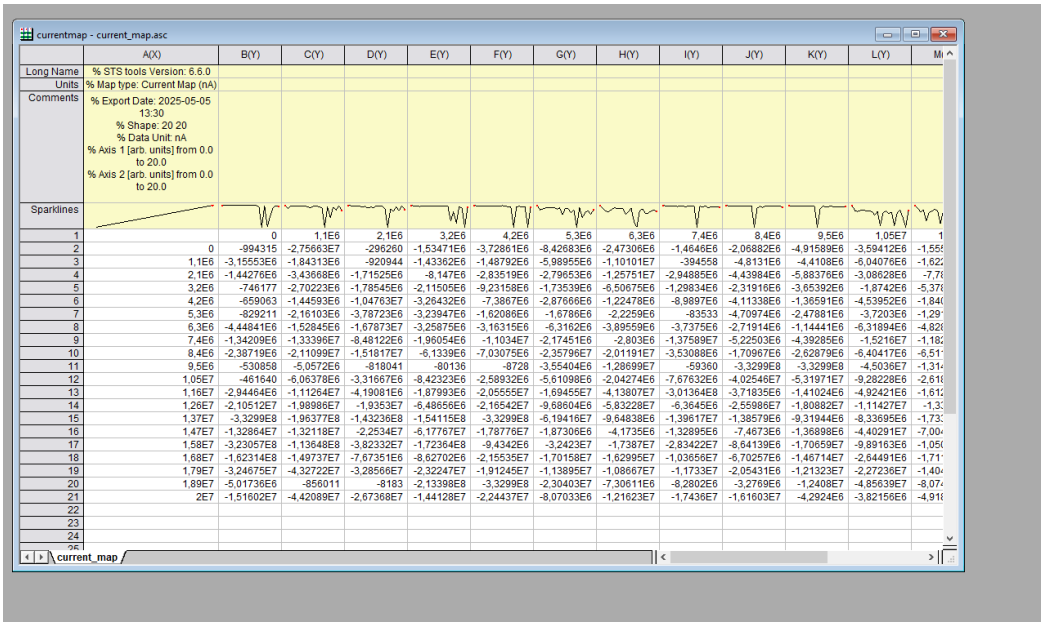
Each file is generated according to the parameters currently selected in the interface. It should appear like this:



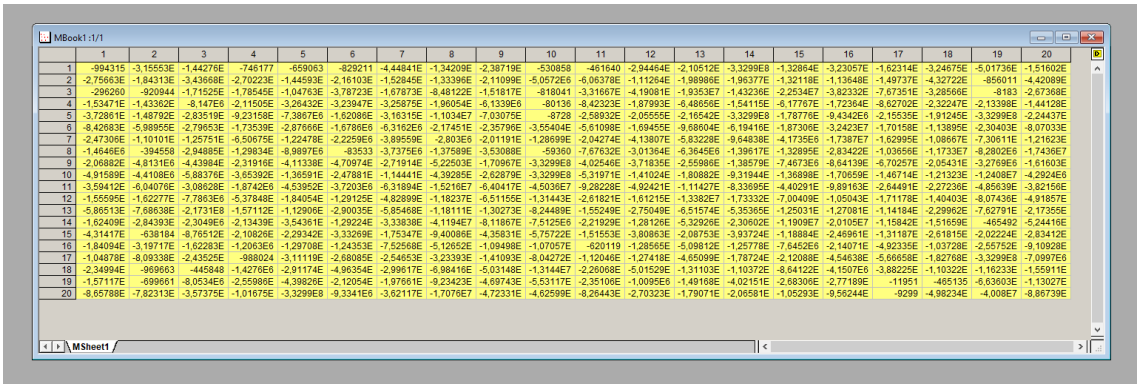
📈 How to Use in Origin

To import and visualize the maps in **Origin**:

1. Go to **File > Import > Single ASCII** and select the .asc file.
2. Origin will create a worksheet with the raw data.



3. Navigate to **Worksheet > Convert to Matrix > Direct** (or **Interpolate**, if needed).



4. Go to **Plot > Contour/Heatmap > Heatmap** to visualize the data.

After these steps, you will see the spatial map rendered in Origin.

File Structure (ASC Format)

Each file follows a structured ASCII format compatible with Origin, containing:

Header Metadata (lines prefixed with %):

- % STS tools Version: — STStools version used to export the file.
- % Map type: — The type of map (e.g., *Current Map (nA)*).
- % Export Date: — The date and time of export.
- % Shape: — Matrix dimensions (e.g., 20 20 for 20 rows × 20 columns).
- % Data Unit: — Units of measurement (e.g., nA).
- % Axis 1 [...] and % Axis 2 [...] — Axis details with ranges and units.

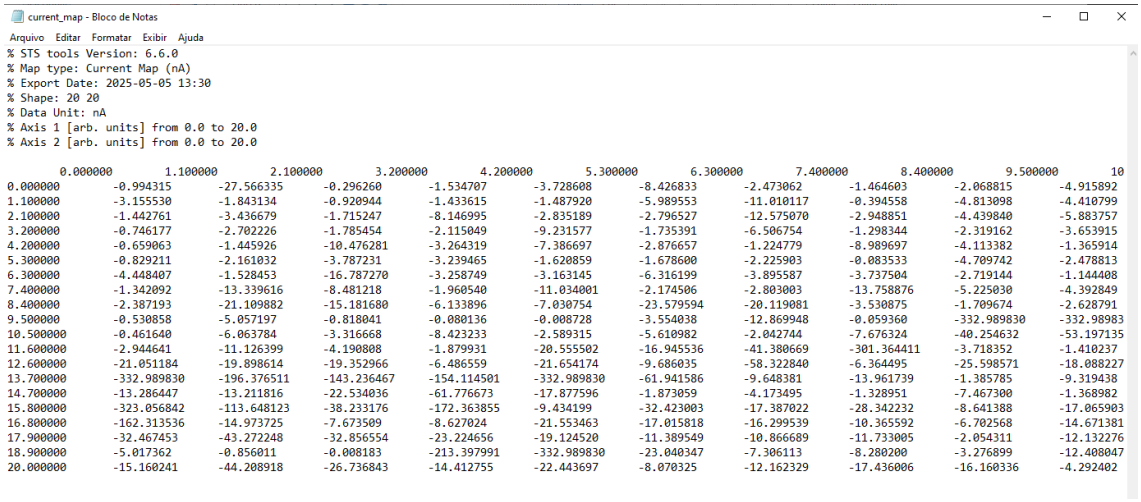
Data Grid Format:

After the header, the data block follows this structure:

\t	X_0	X_1	X_2	...	X_n
Y_0	Z_{00}	Z_{01}	Z_{02}	...	Z_{0n}
Y_1	Z_{10}	Z_{11}	Z_{12}	...	Z_{1n}
...					
Y_m	Z_{m0}	Z_{m1}	Z_{m2}	...	Z_{mn}

- The **first row** (after the header) contains **X-axis values** (typically spatial coordinates or voltage).
- Each **subsequent row** begins with a **Y-axis value**, followed by tab-separated **Z-values**, which represent the measured data (e.g., current or dI/dV).

💡 This format ensures compatibility with visualization tools such as Origin, MATLAB, or custom Python scripts.



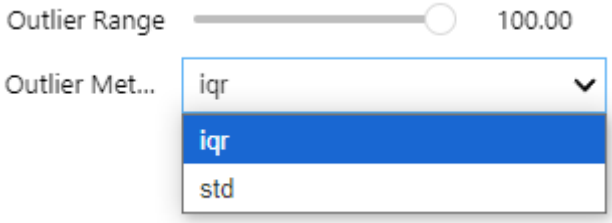
2.3.5 Smoothing

This slider adjusts the **degree of smoothing applied to the I×V curves**, expressed as a **percentage of the curve length**. The default setting is **3%**.

Smoothing is applied before generating maps such as **gap**, **doping**, or **derivative-based** maps. It helps to reduce noise and stabilize numerical differentiation.

⚠ **Note:** Avoid using values **greater than 6%**, as this may excessively flatten the curve and **eliminate important physical features**, leading to inaccurate map calculations.

2.3.6 Outlier range and method



This control improves the **visual quality of the maps** by removing outlier values from the colormap scale.

- The **slider** sets the **intensity limit** as a percentage of the maximum value.

  Default is **100%**, meaning all values up to the maximum are included.

- The **dropdown menu** selects the **method used to detect and exclude outliers**:
  - **iqr** — Uses the **interquartile range** (Q1–Q3) to detect and eliminate outliers.
  - **std** — Uses the formula:

$$mean \pm (Outlier_{Range} \times standard\ deviation)$$

These filters affect only the **visual representation** of the maps (not the underlying data), helping to avoid color saturation and improve contrast in cases where a few extreme values skew the color scale.

💡 **Tip:** Use iqr for datasets with skewed distributions or outliers, and std for more symmetric or normally distributed data.

2.3.7 Color min and max

These two sliders allow you to manually define the **minimum and maximum values** for the **colormap scale** used in map visualization.

By adjusting these limits, you can:

- Improve contrast
- Focus on specific value ranges
- Highlight subtle variations in regions of interest

This setting applies to all displayed maps (e.g., current, derivative, gap, doping), and affects only **visual appearance**, not the underlying data.

💡 **Tip:** Use in combination with the **Outlier Range** setting to fine-tune visual clarity, especially when your data has a wide dynamic range or isolated spikes.

2.3.8 Colormap

This dropdown menu allows you to select the **colormap** used to visualize the data in all map panels.

The colormap defines the **color gradient** applied to represent variations in map values.

- The **default colormap** is **jet**, a widely used gradient ranging from blue to red.
- Other options include perceptually uniform and scientific palettes, such as viridis, plasma, inferno, and more.

🎨 **Tip:** Use perceptually uniform colormaps (e.g., viridis) for publications or when precise interpretation of intensity gradients is important.

2.3.9 Automatic Colormap

Click this button to **enable or disable automatic colormap scaling**.

- When **enabled** (dark green), the colormap limits are automatically set based on the **minimum and maximum values** of the currently displayed map.
- When **disabled** (light green), you can manually control the color scale using the **Color Min** and **Color Max** sliders.

⚠️ **Note:** If your data contains **outliers**, automatic scaling may compress the color range for the majority of values. In such cases, it is recommended to **disable** this function and manually adjust the color scale for better contrast.

2.3.10 V min and V max

These two sliders define the **voltage window** used to calculate the **Integral Map**.

- **V Min**: Sets the **lower limit** of the voltage range.
- **V Max**: Sets the **upper limit**.

The Integral Map is computed by integrating the STS signal (typically the I×V or dI/dV curve) over the selected voltage range at each spatial point.

📌 **Tip:** Use this feature to focus on specific electronic features (e.g., occupied or unoccupied states) by selecting an appropriate energy window.

⚠️ The **V Min** must always be **less than V Max**, or the calculation will not proceed.

2.3.11 Threshold

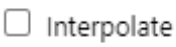
This slider defines the **threshold value** used to calculate the **gap** and **doping maps**, expressed as a **percentage of the maximum value** of the **dI/dV curve** at each point.

🔷 **Example:** If the threshold is set to **5%**, any portion of the dI/dV curve that falls **below 5% of its maximum value** will be considered part of the **energy gap**.

This method helps identify the extent of the gap region in the STS data and assists in determining the doping level based on the asymmetry of the curve.

⚠️ **Note:** Choosing a threshold that is too high may overestimate the gap; values that are too low may fail to detect real features in noisy data.

2.3.12 Interpolate



This option toggles **interpolation** for the visualized maps.

- When **enabled**, the map data is **interpolated** to create a **smoother, higher-resolution visualization**, which can enhance visual interpretation.
- When **disabled**, the map is shown in its **original resolution**, preserving raw pixel values.

ℹ️ **Note:** Interpolation affects **only the display**, not the underlying data or exported files. It is useful for presentations or qualitative analysis but should be disabled when inspecting precise numerical details.

2.3.13 Save Figure

Click this button to **save the current map visualization as a .png image**.

- The image will be saved with the filename **Heatmap_plot.png**.
- It will be stored in the **same directory** where the Jupyter Notebook is currently running.

This feature allows you to quickly export and document the current map for use in reports, presentations, or publications.

💡 **Tip:** Use this to capture a snapshot of the current colormap settings, interpolation, and axis ranges exactly as they appear on screen.

3. Quick User Guide

Installation

To use the STStools library, you must have:

- Python ≥ 3.10 (Anaconda recommended)

- Jupyter Notebook or Visual Studio Code with Jupyter extension

- (Optional) Origin software for visualization

Installation command:

pip install ststools

Loading the Library

In a Jupyter Notebook cell, import the library:

import ststools as sts

After loading, STStools initializes a GUI interface for STS and map analysis.

Loading Data

Use one of two buttons:

- Load STS Files — for individual STS curves

- Load Grid Files — for grid-based maps

If using multiple files, define number of rows/cols.

You can also average multiple curves using the checkbox.

Using the Display Interface

After loading STS data, launch with:

sts.Display()

Key features:

- Curve visualization

- Smoothing and thresholding

- Histogram generation

- Export of STS and derivative data

Using the Map Interface

For grid-based analysis, use:

sts.Map()

Main features:

- Map type selection (Current, Gap, Doping, etc.)

- Position selectors

- Map and STS exports (.asc)

- Smoothing, interpolation, and colormap control

Exporting Data

You can export:

- STS and dI/dV curves (.csv)

- Maps (.asc): current_map, derivative_map, gap_map, doping_map, integral_map

- Fixed-line maps and STS profiles

Example .asc Format

% STS tools Version: 6.6.0

% Map type: Current Map (nA)

% Shape: 20 20

% Axis 1 [nm] from 0.0 to 20.0

% 0.0 1.0 2.0 ... 20.0

% Axis 2 [nm] from 0.0 to 20.0

% 0.0 1.0 2.0 ... 20.0

0.1 0.2 0.3 ...

Importing Maps into Origin

1. File > Import > Single ASCII

2. Origin creates a worksheet

3. Worksheet > Convert to Matrix > Direct

4. Plot > Contour/Heatmap > Heatmap

Supported File Types

Microscope | STS File | GRID File

----------|----------|----------

OMICRON | .txt | .ibw, .txt

NANOSURF | .nid | .nid

NANONIS | .dat | .dat

Contact & Source Code

Author: Rafael Reis Barreto

Email: rafinhareis17@gmail.com

GitHub: github.com/rafinhareis/ststools