

# MIEZEPY, a user guide

Andreas Wendl\*

June 2019

## 1 Introduction

MIEZE is a variation of neutron resonance spin echo, which allows the usage of depolarizing sample environments. For more details, see <sup>1</sup>.

## 2 Installation

The following sections describe the installation process for the different operating systems. See the website for latest documentation and updates: [https://scgmlz.github.io/MIEZEPY\\_website/](https://scgmlz.github.io/MIEZEPY_website/) Currently only installation through the terminal/command prompt is supported. Note that at this point the installation of MIEZEPY requires basic knowledge in terminal input. In the following steps we will install python and the required dependencies to run the package both in scripting and GUI mode.

### 2.1 Linux

Linux is the recommended operating system, since it was mainly developed and tested there.

The GitHub repository of the MIEZEPY project can be downloaded [https://github.com/scgmlz/NSE\\_Soft](https://github.com/scgmlz/NSE_Soft). As the download completes the user is required to unzip the .zip content and open a terminal window in the downloaded folder. This can be done by either rightclicking an empty field within the folder and selecting “Open terminal” or directly entering “cd Path/to/folder” in an already opened terminal window. The MIEZEPY package has been written exclusively in python and supports only python 3. Therefore an installation of python 3 is required to run the software. If not already present it is possible to install it through the following command:

```
$ sudo apt-get install python3.7
```

Furthermore, we require to install a common package manager for python called pip through:

```
$ sudo apt-get install python3-pip
```

The last step might not be necessary as for some python versions pip is included by default. If the installation is successful we can proceed by checking the installation versions:

```
$ python3 --version
$ pip3 --version
```

---

\*Physik Department TUM, 85748 Garching, andreas.wendl@ph.tum.de

<sup>1</sup>MIEZEpy Paper, to be published

If the above command fails try to replace pip3 with pip as pip not being installed by default on linux it might select the former namespace. Finally, the output should be the version numbers of python and pip above 3.6.6 and 18.1 (at the time of writing) respectively. The MIEZEPY package has dependencies on some common (like numpy) and less common (like PyQt) python libraries. These need to be installed in order to launch the package. In any terminal window enter the following commands:

```
$ sudo pip3 install -r requirements.txt
$ sudo pip3 install git+git://github.com/pyqtgraph/pyqtgraph.git
$ sudo pip3 install git+git://github.com/AlexanderSchober/simpleplot_qt.git
```

The first command installs the requirements defined in the requirements.txt file included in the repository. The second line installs a graphical PyQt library and the third command installs the plotting library derived from PyQt and pyqtgraph. Finally the software can be installed through the command:

```
$ python3 setup.py install
```

Once the installation is finished the software can be tested by launching the python interpreter through:

```
$ python3
>> import miezepy
>> miezepy.__version__
```

This should provide the version number of the downloaded distribution.

## 2.2 Windows

Since the program is based on python, it also works under windows. Nonetheless, the installation can be a little more complicated.

### 2.2.1 Preparations

It is recommended to install python via the Anaconda package. A Win10 version can be found here: <https://www.anaconda.com/download/#windows>. After you have installed Anaconda, we recommend to create a separate environment within anaconda solely for MEIEZPY. This ensures, that all packages for MIEZEPY are kept together and no dependencies are overwritten, when installing other python software. The full instructions for environments can be found here: <https://conda.io/docs/user-guide/tasks/manage-environments.html>

First, open the Anaconda Prompt, which is also the recommended command line on windows for interacting with anaconda/python.

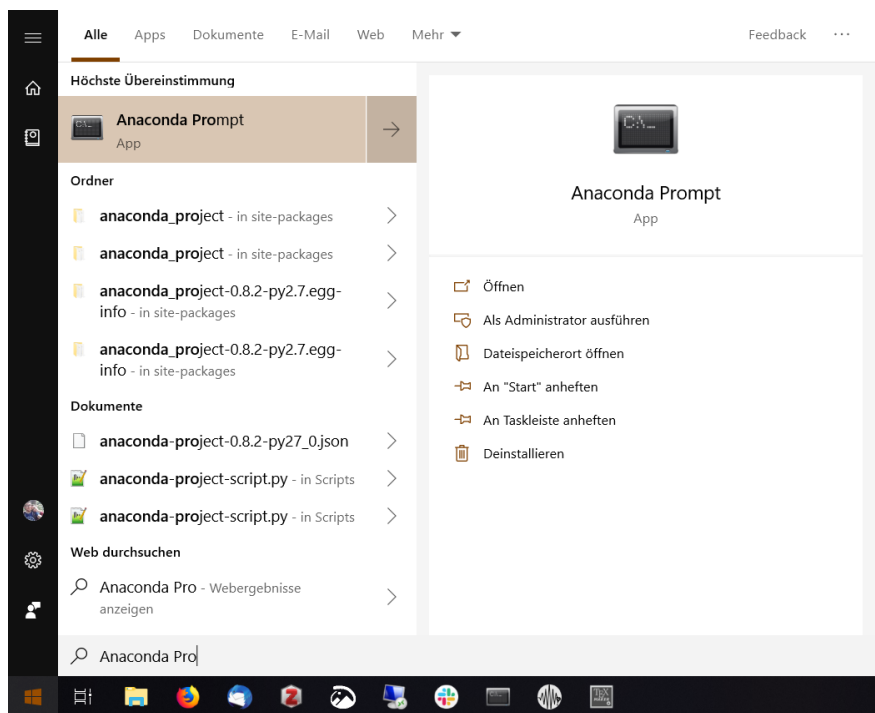


Figure 1: Caption

A new environment named "MIEZEPY", with python 3.7 and the required modules/packages can be created with the command

```
conda create -n MIEZEPY python=3.7 numpy matplotlib scipy iminuit pillow cython imageio pyopengl
```

The corresponding version of python and the requirements will be installed and stored under "user/Anaconda2/envs/MIEZEPY". You can always check the packages of an environment by opening the environment

```
activate MIEZEPY
```

and then displaying the installed packages via:



- Check installation via Anaconda:

```
conda list
```

The list of all packages should be displayed. You should find the current version of MIEZEPY and `simpleplot_qt`. If you have installed a new version of MIEZEPY on top of an already existing, check here, that the old version was deleted/overwritten properly.

The installed software can now be tested. (since there is a slight bug in version 0.2.0, where one folder is not correctly created upon installation, open the MIEZEPY from its downloaded folder:

```
cd Path\to\MIEZEPY\folder
```

```
activate MIEZEPY
```

```
python -m mieze.py.mieze
```

The GUI should open, which is explained in the following section:

### 3 User Interface

Right now, there is only the option to start the program via the command line. Open your command line, for example the Anaconda-Bash. Activate your python3 environment.

```
activate MEIEZPY
```

Run Python and execute

```
$ from miezepy.mieze import Mieze  
$ Mieze(True)
```

After the GUI started, you can choose between the following pages, which are arranged in the order of their typical usage during data reduction: environment handling, data loading, mask creation, data reduction, result plotting and saving and loading sessions.

#### 3.1 Environments

The first page allows to create and organize the environments, which are currently opened. The current environment is highlighted and can be seen in brackets after MIEZEPY in the title of program window. The +/- Buttons (1) allow to create and remove the environments. A proper name (2) should be selected. All the following pages can be also directly accessed through the corresponding buttons (3). The "Masks"-Button has a special feature: By drag-and-dropping it onto another environment, you can copy all masks to the other environment.

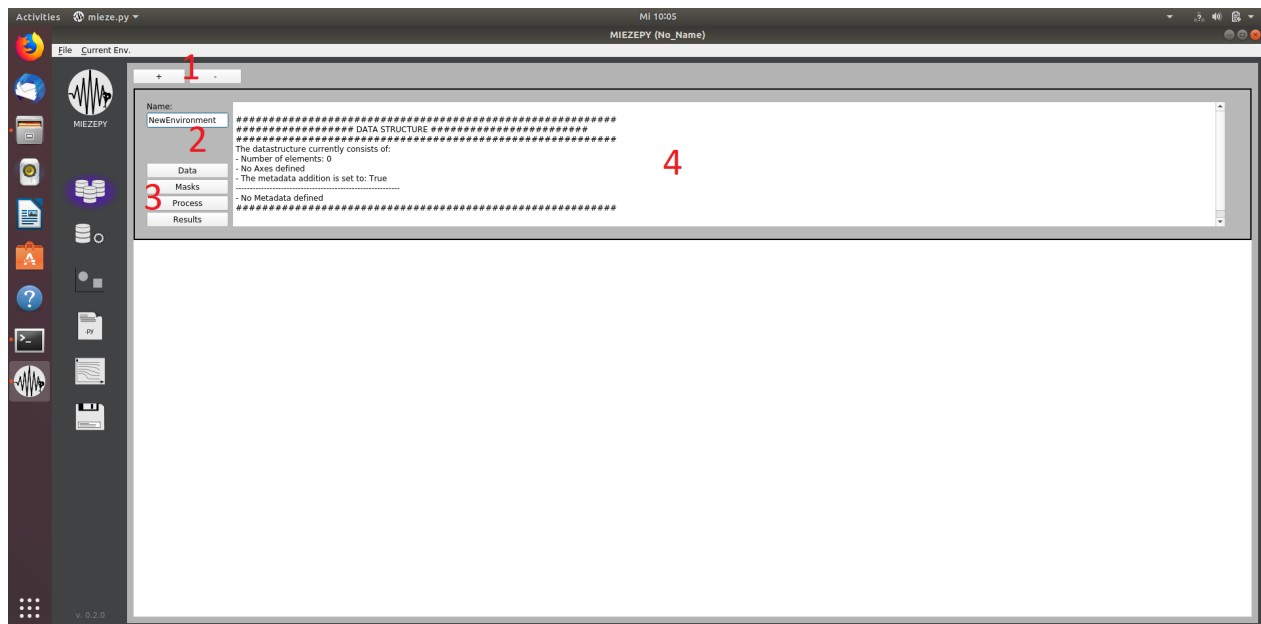


Figure 3: The Environment tab: 1: create a new environment 2: name selection 3: switch to the other tabs (data import, mask creation, data reduction and results) 4: overview of the environment parameters

## 3.2 Data Import

The import of .TOF files, which are provided by the NICOS instrument control software is handled on this page. The GUI is shown in Fig. 4. Again, with the +/- key (1), a new dataset can be added or removed. All datasets of this environment are listed in (2). Every dataset contains the following information:

- parameter: Name your dataset to describe the investigated parameter in an appropriate manner.
- measurement: If you have measured the same parameter (Q, T, B ...) several times, you may want to specify, that by giving it different numbers.
- reference: Select, if this dataset is your reference/resolution measurement.
- background: Select, if this dataset is your background measurement.
- visualize (4D): Visualize the detector images of the various foils and time channels in the dataset.
- The table summarizes all the metadata of the different TOF files. One TOF file is measured at a specific set of first frequency, second frequency, sample-detector distance (LSD) and wavelength. (Additionally, the metadata contain the monitor counts, proportional to the current neutron flux). Therefore, one TOF file corresponds to one row, where the "Echo" (short for spin echo time) is calculated from these four parameters. The "Echo" therefore serves as an index for matching same conditions from different datasets later-on). For each TOF-file
  - cbox\_0a\_fg\_freq\_value: frequency of the first resonant flipper as read from the TOF file
  - cbox\_0b\_fg\_freq\_value: frequency of the second resonant flipper as read from the TOF file
  - psd\_distance\_value: sample-detector distance as read from the TOF file
  - selector\_lambda\_value: wavelength as read from the TOF file
  - monitor1: monitor counts as read from the TOF file
  - Echo: Mieke time / spin echo time, as calculated from the following four parameters:
  - Freq. First: manipulated value of the first frequency for calculation
  - Freq. Second: manipulated value of the second frequency for calculation
  - lsd: manipulated value of the sample-detector distance for calculation
  - Wavelength: manipulated value of the wavelength for calculation
  - Monitor: manipulated monitor counts

are listed. These parameters can be manipulated in (5).

A list of the .TOF files, belonging to one dataset can be selected under "Files" (3). The current TOF-file is visualized in (4) to check your selection before loading. With the "Populate this dataset" - button, the data from the selected TOF-file, as well as the manipulations to the metadata can be loaded into the selected dataset. The metadata can be manipulated by writing values in the lines (Fig. 4 5).

- A single value will overwrite ALL entries for this metadata for this dataset.
- A comma-separated list can be used to change each entry separately (in the order, they appear in the table). A comma separated list, which is shorter than the number of entries, will only change the entries covered by the list - remaining entries are not changed.

Note: This will NOT overwrite the .TOF files. Thus, your original data remain the same.

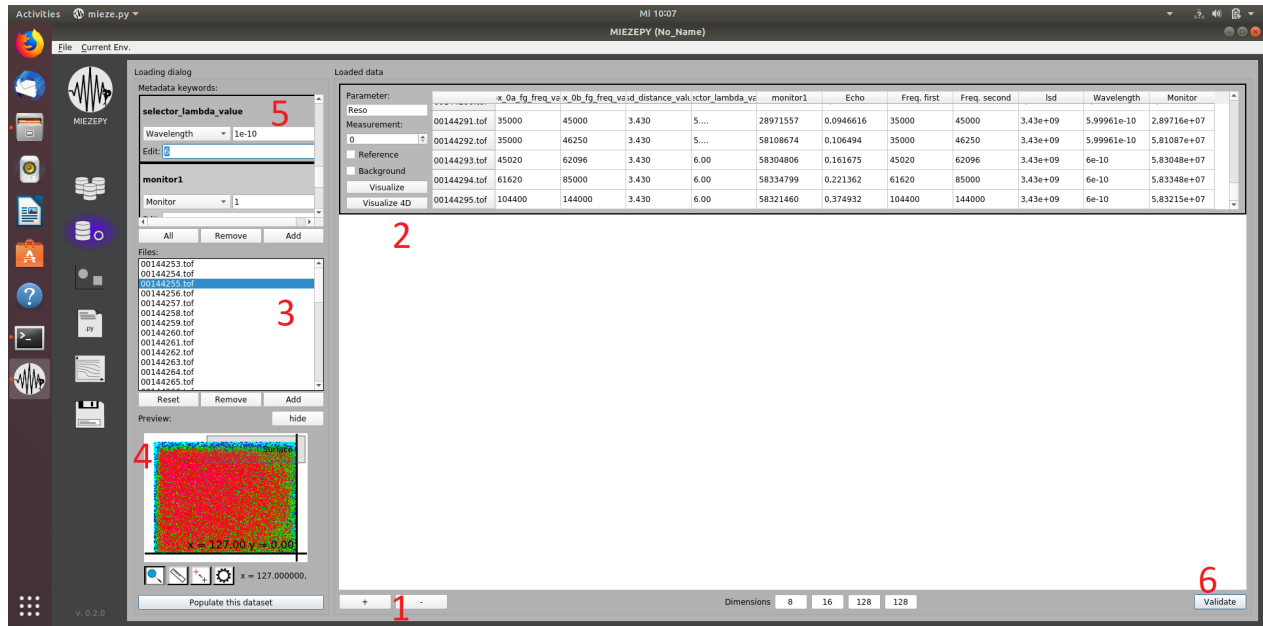


Figure 4: The Data Import tab: 1: create a new dataset 2: List of datasets with their information 3: TOF-files to be imported into an environment 4: visualization of the integrated intensity in the selected TOF file 5: manipulation of the metadata of the selected dataset 6: validate - checks if input is valid and loads it into the environment.

### 3.3 Masks

An integral part to every data reduction is to select, where the interesting signal appears on the position sensitive detector. The mask creation tab is shown in figure 5. To create a new mask (1) select "New..." from the drop-down menu and write the name. Then confirm with "Enter". New elements can be added to the mask with the +/- key (2). The various options (3) are the following:

- Rectangle (a)
  - Position: x and y coordinates in pixels
  - Angle: rotation of the element in degree
  - Dimensions: width and height in pixels
- Arc (b)
  - Position: x and y coordinates
  - Angle: rotation of the element in degree
  - Angular range: upper and lower angular boundary of the circle segment (0 degree = parallel to x-axis)
  - Radial range: inner and outer diameter of the circle segment in pixels
- Triangle (c)
  - Position: x and y coordinates
  - Angle: rotation of the element in degree



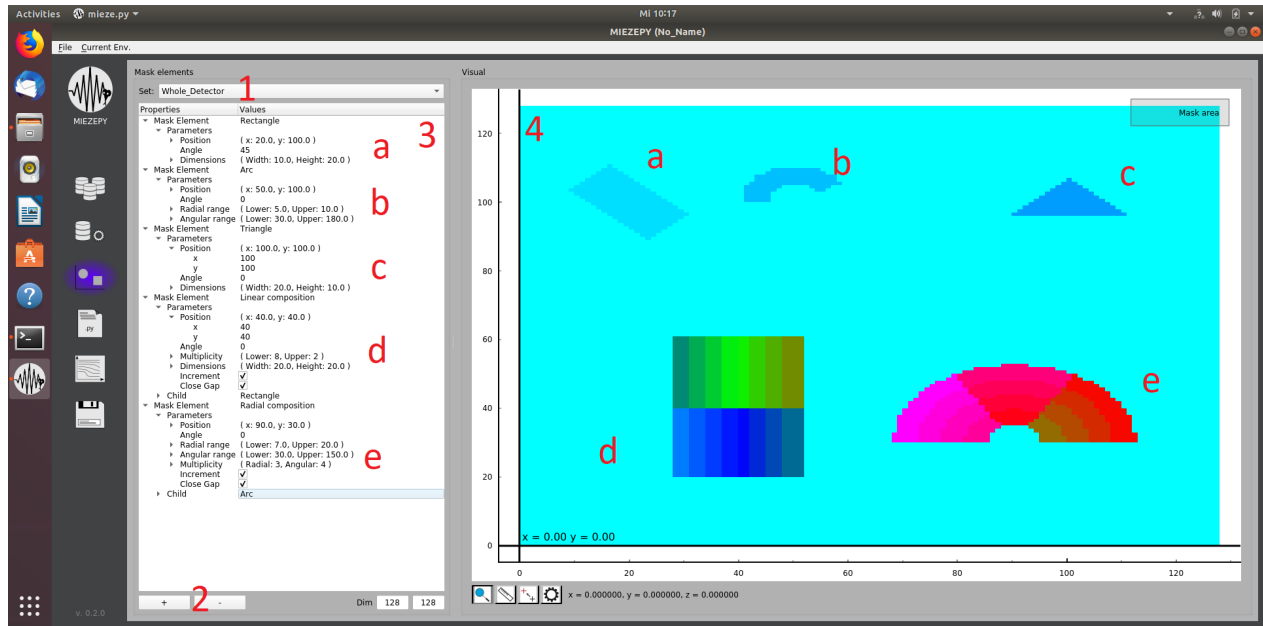


Figure 5: The Mask Creation tab: 1: selection of the current mask, when "New..." is selected, you can type the new name in here and confirm with the "Enter"-key. 2: Add a new shape to the current mask. 3: Manipulate the shapes within this mask. 4: Visualization of the current shapes.

- Dimensions: width and height in pixels
- Linear composition (d)
  - Position: x and y coordinates
  - Angle: rotation of the element in degree
  - Dimensions: width and height in pixels
  - Multiplicity: number of child elements in x (lower) and y (upper) direction
  - Increment: If selected, each individual child will have an own value, making it possible to use the phase correction with the mask method.
  - Close gap: If selected, the dimensions of the child (if rectangular) will be selected, so that no gaps remain between the individual child elements. Overwrites child properties.
  - Child: The child element, which should be multiplied, can be selected and manipulated, as described in (a-c)
- Radial composition (e)
  - Position: x and y coordinates
  - Angle: rotation of the element in degree
  - Angular range: upper and lower angular boundary of the circle segment (0 degree = parallel to x-axis)
  - Radial range: inner and outer diameter of the circle segment in pixels
  - Multiplicity: number of child elements in radial and angular direction

- Increment: If selected, each individual child will have an own value, making it possible to use the phase correction with the mask method.
- Close gap: If selected, the dimensions of the child (if arc) will be selected, so that no gaps remain between the individual child elements. Overwrites child properties.
- Child: The child element, which should be multiplied, can be selected and manipulated, as described in (a-c)

Note: If you want to copy masks between environments, drag-and-drop the "masks" button in the environments tab (figure 3). Furthermore, you can save and load masks under "Current Env."

## 3.4 Data reduction

### 3.4.1 Process

The data reduction page is shown in figure 6. There are three windows under this tab, namely the Process (1), Panel (2) and Scripts (3) window. On the Process window, you can select all main properties of the data reduction. For each echo within the dataset, you can select the foils, which should be analyzed (4). The datasets are listed in (5). You can select, which timechannels should be considered in the contrast fitting (6). Usually, you want to select all timechannels, unless detector-issues require else.

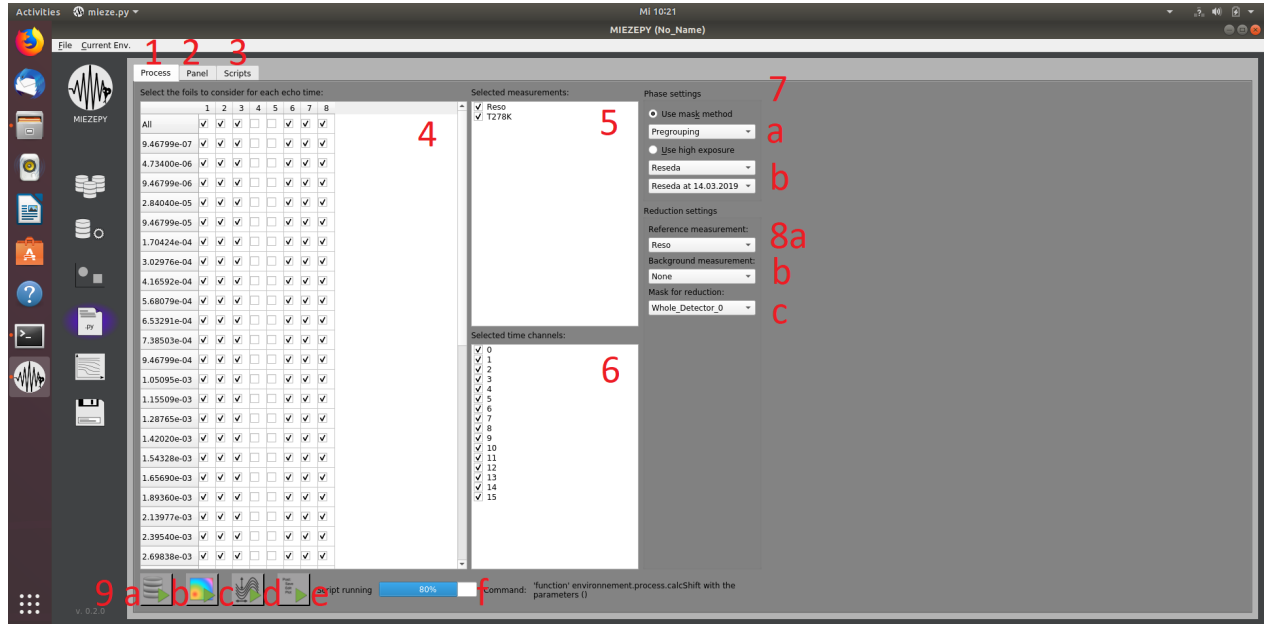


Figure 6: Overview of the data reduction tab. For details, see text.

There are two options to correct phase shifts on the position sensitive detector (7): (a) The mask method takes the resolution measurement and fits the phase for every tile with different number (see. linear and radial component) within a given mask. For this method, the resolution function needs to have sufficient statistics, so that every can be fitted. The phase for each tile can then be translated in a shift of timechannels, so that all tiles are in phase. For the whole dataset, this correction is then applied. If the detector configuration is known, it is possible to calculate the phase correction from the metadata (b). For this method, one preemptive measurement with high exposure with the detector needs to be made, to obtain information about foil warping. The resolution and data measurements may therefore be taken with much shorter exposure, since the phase information does not need to be collected.

After the raw data were corrected, further data reduction is required before the intermediate scattering function  $S(q, \tau)/S(q, 0)$  is obtained (8). As all spin-echo methods, also MIEZE requires normation of the data with a resolution measurement of an elastic scatterer (a). Depending on the experiment, it may be necessary to subtract a background measurement (b). The interesting parts of the position sensitive detector can be selected under "Mask for reduction", where you can select any mask you have created earlier (figure 5).

Every step of the data reduction is performed of an own script. They can be executed at the bottom of this tab (9): (a) The data are loaded into python. (b) The phase correction is performed. Caution: This step can take some time! For the example displayed in the figure, it took approximately 7 minutes. (36 Tiles, 43 datapoints, 2 measurements, 1 core used). (c) The background is subtracted and the data divided by the resolution measurement. (d) Additional commands can be passed in the post-process script. (e) A progress bar informs about the status of the current calculation. (f) (Error) messages are displayed.

### 3.4.2 Panel

The Panel tab can be used to visualize and investigate your raw data in detail. It is shown in figure 7. (1) Selection of the data set. Since some calculations in the background can be demanding, you can either compute the result with the button, or use the "live" mode, where it is calculated automatically after each change. (2) Selection of the mask. You can manipulate the mask here, in the same way as in the mask page. (3) The whole detector image. (4) The detector image with the mask applied. (5) The fit of the time channels. (6) The raw non-normalized contrast.

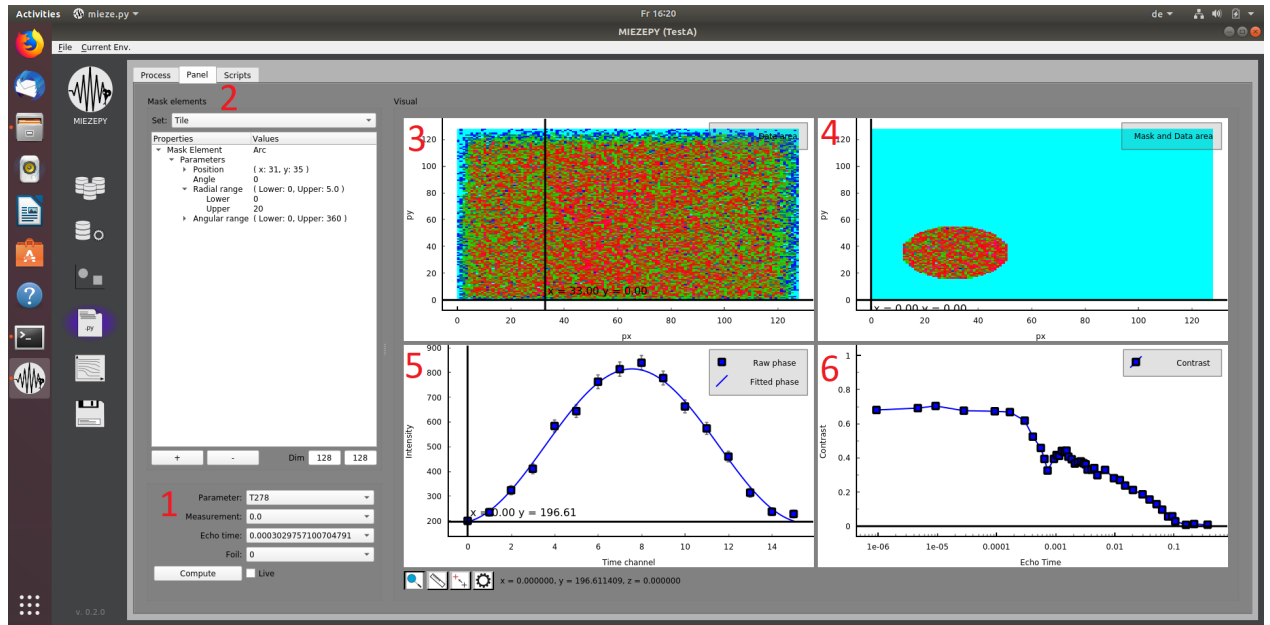


Figure 7: The Panel tab, see text for details.

### 3.4.3 Scripts

The python scripts, performing the data reduction can all be viewed and manipulated. Usually, the scripts are filled automatically using the Process tab and do not need to be edited. However, there are some cases, where it might be interesting:

- In the fit parameters tab, the number of available processors of your system can be changed. This allows to speed up the calculations due to parallel processing.
- The post reduction tab can be used to run your own python scripts and perform any operation with the data, as you like.

All scripts can be run by clicking the button in the bottom right corner. The GUI button runs the script and opens the process tab, allowing you directly to follow the running script with the progress bar.

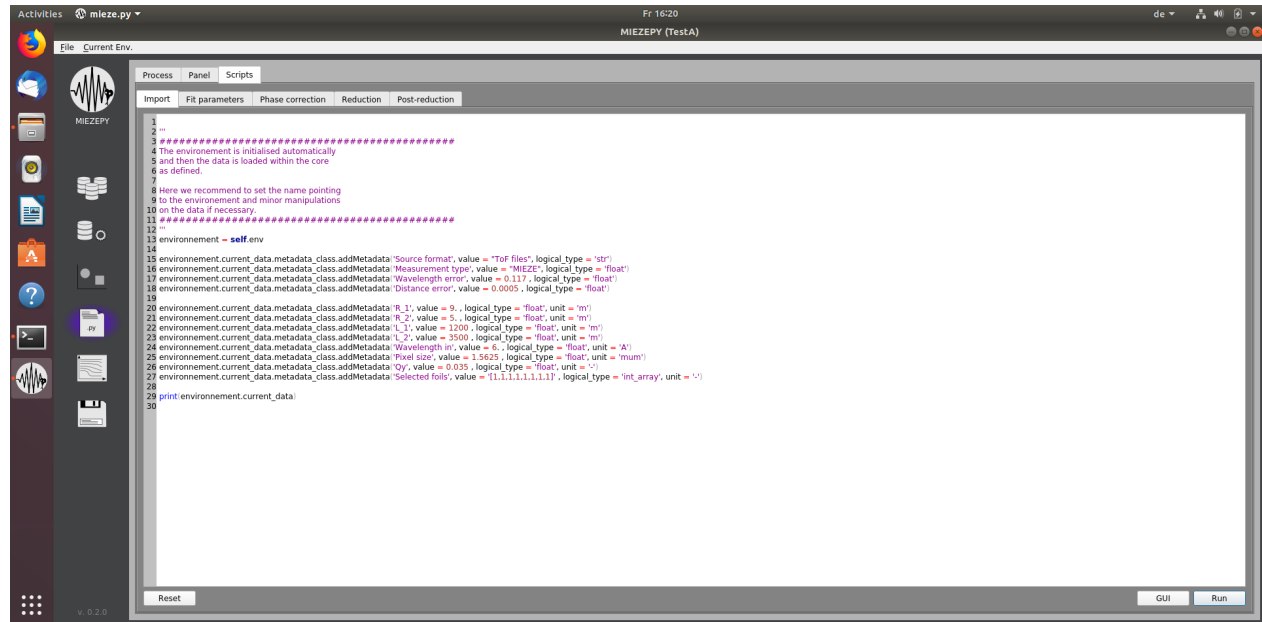


Figure 8: The import script. Parameters of the experiment can be adjusted here.

### 3.5 Results/Plotting

This window allows to plot and export the data, see figure 13. All datasets from the environments are listed in the "data" list (1) and can be updated with the refresh button, after a new calculation was performed. You can drag-and-drop the interesting data to the "plot items" list (2). Single Elements can be removed with the "-" button or the whole list reseted. The properties of each plotted element can be manipulated below (3) under "settings". "Data" shows a table of the data. The plot is shown on the right (4) and updated with the "plot" button in the bottom right corner.

The option to export the content of the graf can be found in the plot settings (gear symbol).

### 3.6 Save/Load

## 4 Work-flow

### 4.1 Organize your data

At first, you need to organize your data, so that the data import is simple. Typically you want to sort by sample, wavelength, momentum transfer, temperature and magnetic field.

After this preliminary work, you can open MIEZEPY by opening the shell/Anaconda Prompt and type:

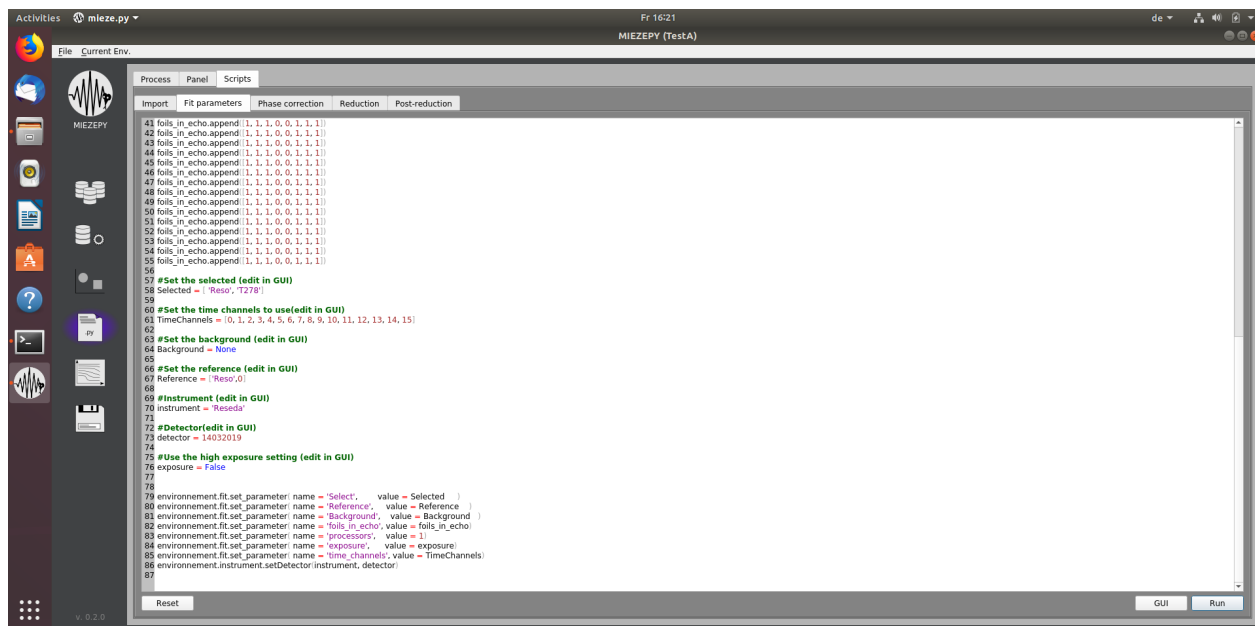


Figure 9: The fit parameters script. Summarizes the parameters for the data reduction.

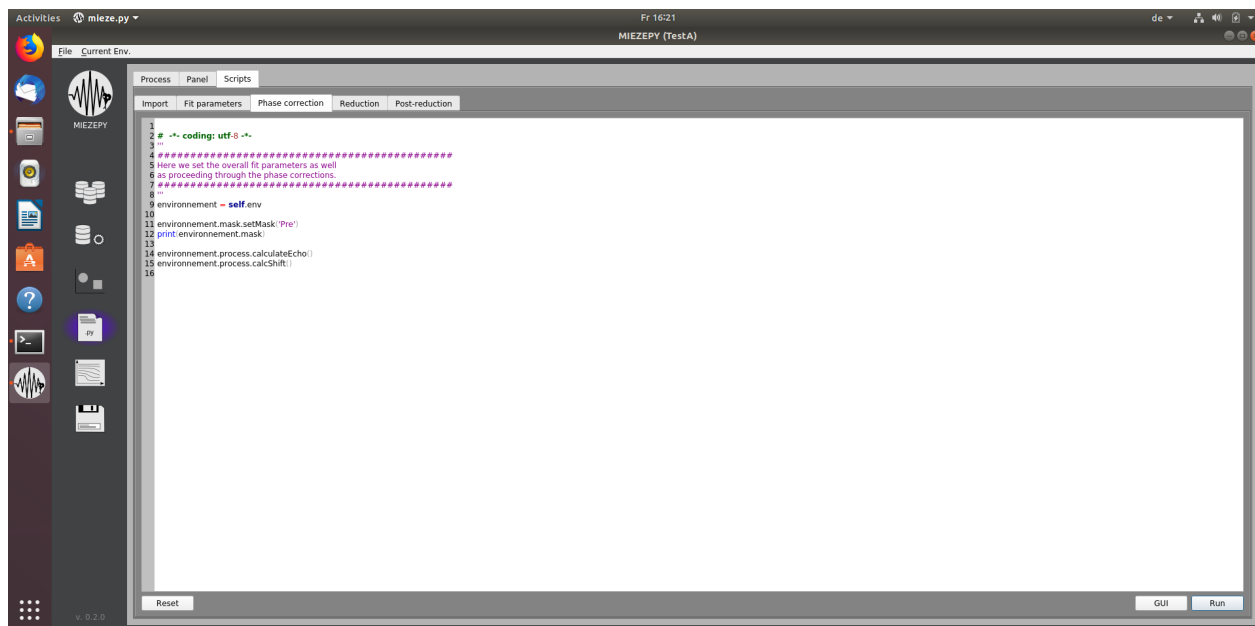


Figure 10: The phase correction tab.

```
$ python -m mieze.py.mieze
```

Create an environment.

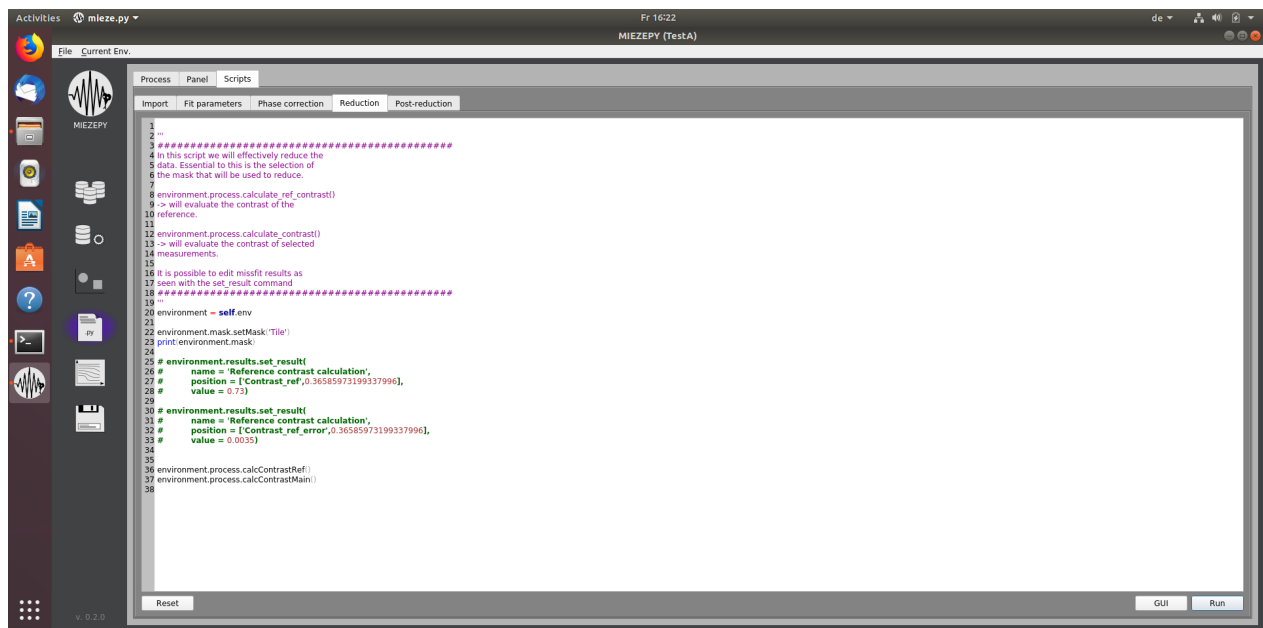


Figure 11: The reduction tab.

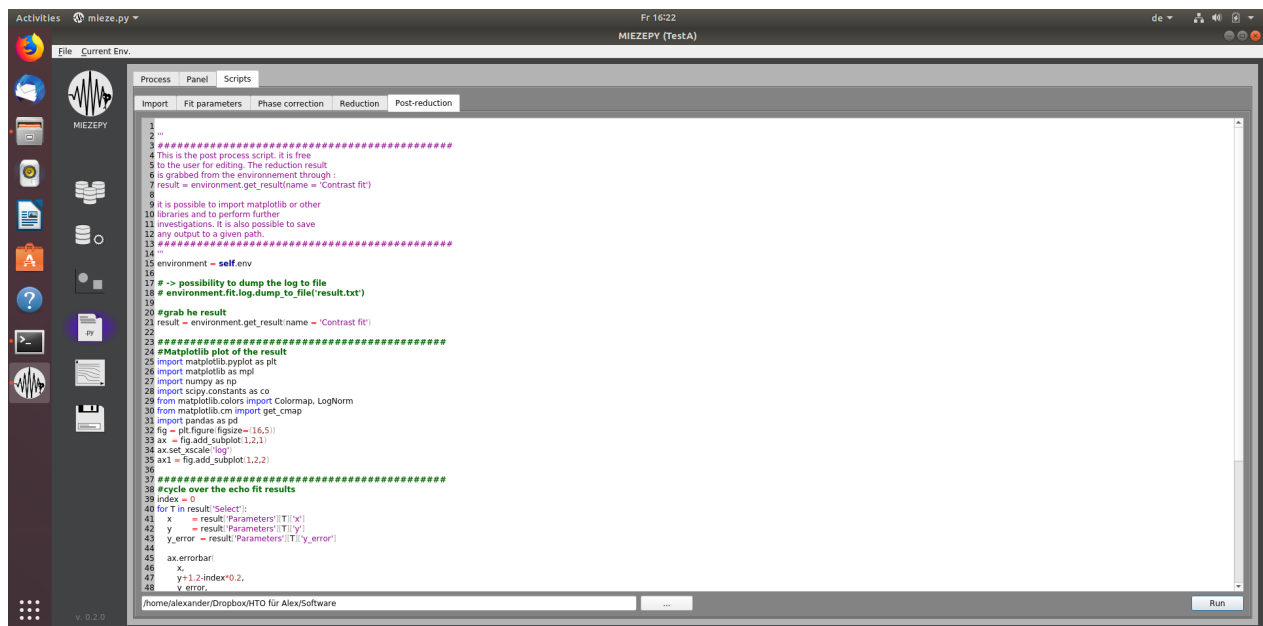


Figure 12: The post reduction tab. In the window at the bottom, you can enter the path, where this script puts output files.

## 4.2 Import data

Now you want to fill the environment with the suitable data. Go to the data import tab and create a new dataset, typically the resolution measurement. Fill it with the corresponding data, which you have sorted in

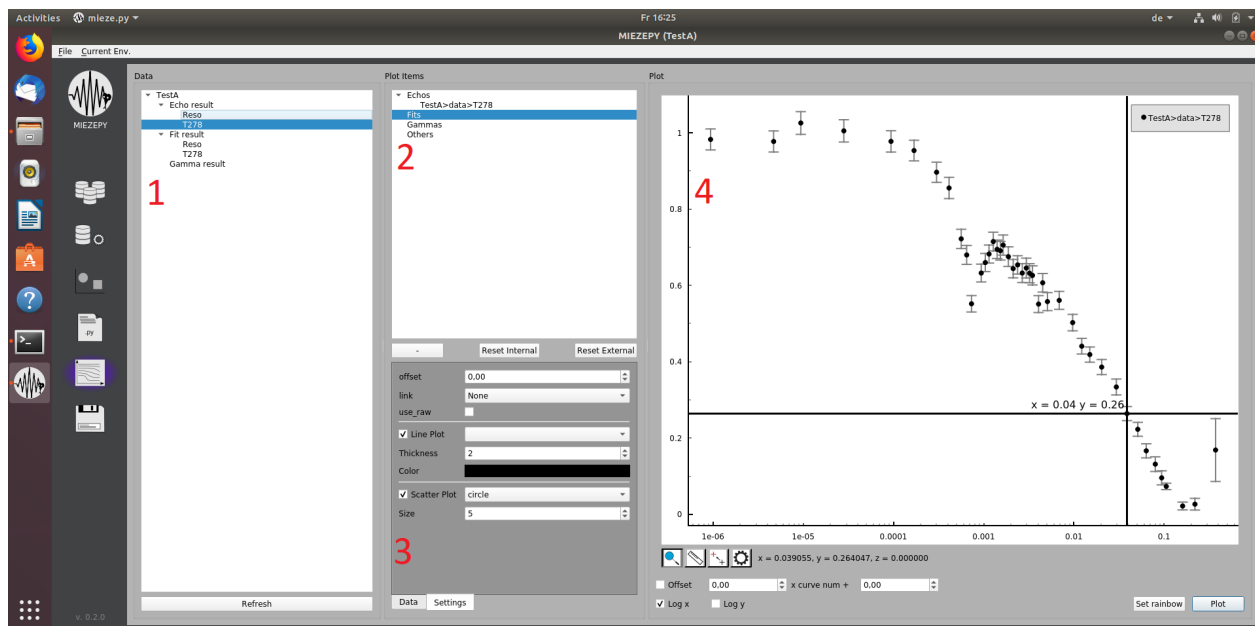


Figure 13: Caption

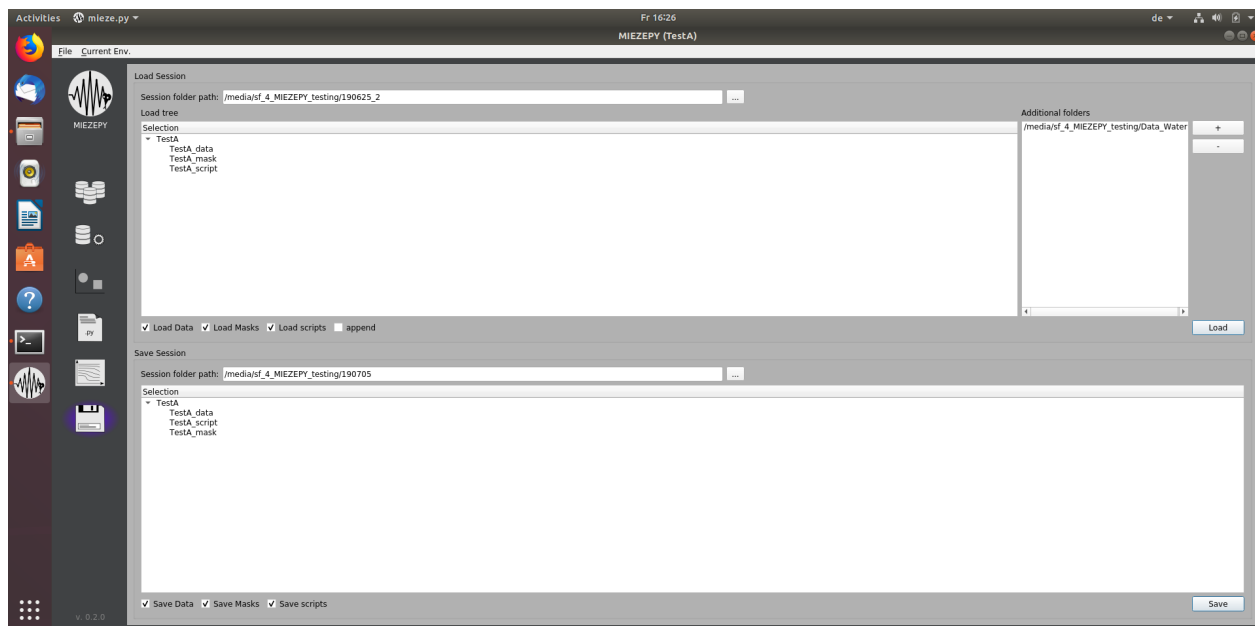


Figure 14: Caption

the first step. Go through the table and identify any .tof files, where wrong metadata were loaded (A typical example would be, that the wavelength is 5.997 instead of 6, due to rounding issues in the data-creation). You can then adjust such errors within the metadata window, as explained earlier. When all datasets are ready, validate the data. It is recommended to save the data at this point.

- 4.3 Create/load masks
- 4.4 Data reduction
- 4.5 Result plotting/export