Manual for the *GUI* of *ANACONDA2*

Benjamin Bolling,[1*]

January 24, 2018

[1*]Department of Physics, Lund University, Lund, Sweden.

Acknowledgements

I want to thank my supervisors Mathieu Gisselbrecht and Bart Oostenrijk for being available for questions, discussions and supervisions as well as the big support both of them gave me while doing this project work. Without them, this work would not have been possible.

Document introduction

The work with developing the ANACONDA2 GUI began as being the main part of my bachelor thesis in physics, see ref. [5], at the Physics Department of Lund University. This project work, also carried out at the Physics Department at Lund University, enabled me to continue develop the ANACONDA2 GUI into what it is today, together with this manual. The project work was registered as Applied Work, which currently has course code FYSB06.

Abstract

This manual will teach the user the basics of data treatment by using the GUI (Graphical User Interface) of the ANACONDA2 (ANAlysis of COiNcidence DAta) data treatment package for MATLAB. ANACONDA2 takes the raw 3D experimental data (t (time), x and y (horizontal and. vertical detector image, resp.)). The reason to have a GUI is so that the user does not have to execute all data treatment procedures via a kernel, which means that new users have to learn the kernel syntaxes and hence complicates more advanced procedures.

The GUI is divided into different tabs for different levels of data treatments, and hence, the manual is divided into the corresponding sections:

File Handler

The file handler section shows how raw experimental data files are loaded and handled by the ANACONDA2 GUI's file-handler and the different operations that can be performed in this tab.

Data Calibration

Raw data has to be calibrated using different calibration procedures for different types of experiments.

Data Conditions and Filters

A data filter can be seen as an obstacle, through which data which fulfills the criteria of the filter can pass. This is done in order to filter out possible noise and uninteresting date. A condition is the term used for the individual conditions (or requirements) that have to be fulfilled before data has passed through, whilst the term filter is referred to for one or for a set of conditions that data has passed through.

Plotting

Plotting is crucial for all experiments in order to comprehend experimental data. There are various alternatives for plotting: Pre-defined and user-defined plot configurations, and the latter either with pre-defined or user-defined signal configurations.

Contents

1 Introduction to the ANACONDA2 GUI

Aim of ANACONDA2, the GUI and this document

The aim of this document is to introduce the reader to the data treatment software ANACONDA2's graphical user interface (GUI).

Central concepts and Definitions

In the ANACONDA2 package, there are two layers for the users: A command-line (kernel-only) layer and a graphical layer. The graphical layer is built upon the command-line layer in order to enable users with a lack of coding (or MATLAB) experience to use the ANACONDA2 package. For the documentation of the kernel-only layer of the ANACONDA2 package, see ref. [1].

Metadata is, as described and defined by ref. [1, chapter 2, pp. 4–6], a set of information defined for each experiment which is "needed to correct, convert, fit and visualize the data". The description of the metadata structure can be found in ref. [1, chapter 2, pp. 4–5], and will not be elaborated any further in this document, since the aim of this document is, as previously described, to describe how to use the ANACONDA2 package graphically via the ANACONDA2 GUI.

A concept that is very central to the ANACONDA2 GUI is data filters. A data filter is built upon one or a set of logical condition(s).

The nature of the experiments that can be analyzed with ANACONDA2 have what is referred to as events with one or multiple hit(s) associated with each events. An event can be e.g. the dissociation of a molecule, whilst the hits are the detected fragments associated with the event.

A signal is defined as something that for every hit has a value, and it could be either raw (coming directly from a spectrometer) or have a physical quantity (e.g. mass-over-charge, kinetic energy, etc.).

System requirements

- A computer which can run the latest* version of MATLAB.
- A licensed and verified version of MATLAB.
- The ANACONDA2 package.

*R2017a is required in this instance. Since ANACONDA2 is and will be continuously updated with new functionalities implemented from updated versions of MATLAB, the latest running version of MATLAB and ANACONDA2 is recommended in order to avoid bugs and compatibility issues.

Getting started

1. Download the ANACONDA2 package to the preferred folder.
2. Change to or add this folder to MATLAB path.[1].
3. In order to start the GUI; write [GUI.main] and press enter.

[1]There are two ways to change/add a folder to the MATLAB path: Graphically or by using commands. By using commands; type [addpath 'dirname'] and press enter (replace 'dirname' with the path of the package, e.g.: /home/user/matlab on Mac OS X). Graphically; browse into the location of the ANACONDA2 package, open the ANACONDA2 folder, right-click on the 'package' folder and select 'Add to Path' → 'Selected Folders'. Note: Do not select 'Selected Folders and Subfolders'!

ANACONDA2 GUI overview

When launching the ANACONDA2 GUI, the screen will look similar to Figure 1. There are four tabs, each representing a different part of the data treatment process in a chronological order.

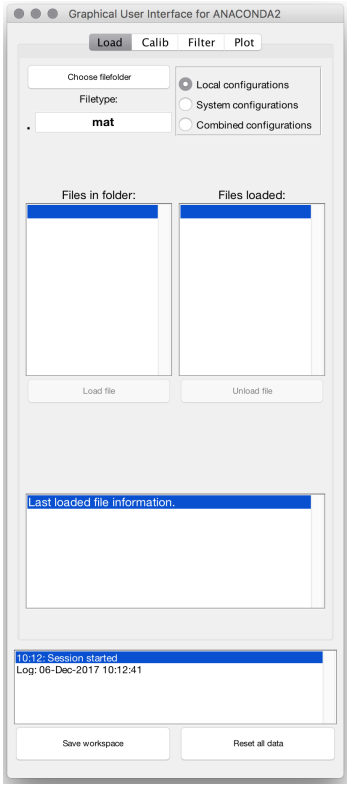

Figure 1: The start screen when launching the ANACONDA2 GUI.

The first tab is a file handler tab, labelled as 'Load' since this is where files are loaded (resp. unloaded).

The second tab is a calibration tab, labelled as 'Calib', where raw data files coming directly from experiments can be calibrated via multiple calibration procedures.

The third tab is a filter tab in which data filters can be constructed, edited, removed and combined.

The fourth tab is the plot tab, in which data can be plotted and where new plot- and signal configurations can be created.

On the bottom, there is a big window which is the 'log box'. In it, the different actions performed with the GUI can be found.

Reset all data means that the GUI will restart and all work will be deleted.

Save workspace means that all files that have been loaded will have their respective metadata files replaced or stored in their respective location.

2 Data treatment part 1: File Handler

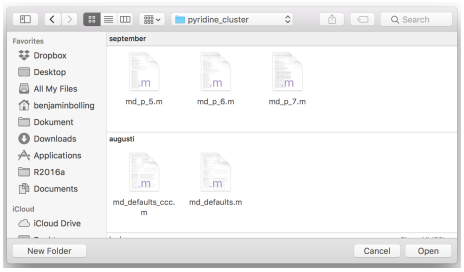

Figure 2: Browse into the folder where the experimental data files are located.

To load data files, they have to be in one of the following filetypes: .dlt, .mat.

In order to get the files from an experiment, click on 'Choose filefolder' and browse into the folder where the experimental data files are located and press 'Open'. Once opened, the filehandler will show all files with the selected file-extension (e.g. '.mat' as seen in Figure 3) in the folder. The filetype can be edited by simply clicking in the box and writing another file-extension, e.g. '.dlt', and pressing enter or with the mouse anywhere outside the box. This will result in that all '.dlt'-files in the folder will be shown.
Before loading a file, the load configurations should be selected which describes how a file is to be loaded.

- 'Local configurations' selected means that if there is a readable configuration file in the same folder with the name 'md_[filename].m', this file will be used when the [filename] data file is loaded.
- 'System configurations' means that a pre-defined spectrometer can be selected in order to generate the metadata necessary for the data treatment.
- 'Combined configurations' means that both local and system configurations will be used. If any overlap is found for a metadata, the metadata defined in local configurations will be used.

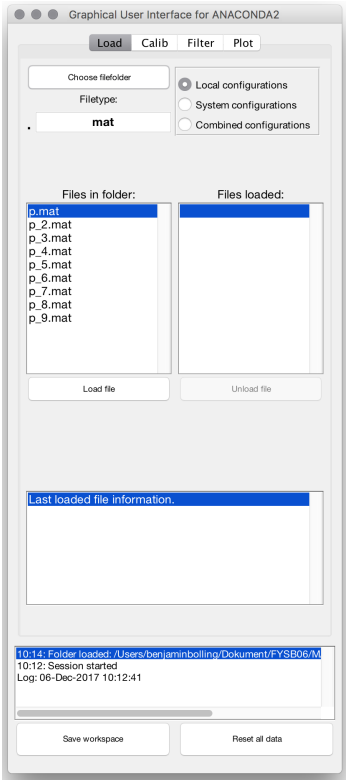

Figure 3: Files in selected folders with selected filetype can be seen.

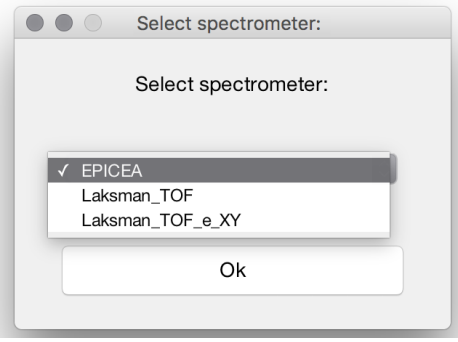

Figure 4: When using system configurations to load a file, there are three spectrometers that can be chosen from.

There are currently three available spectrometers: EPICEA, Laksman_TOF and Laksman_TOF_e_XY. Their properties and differences can be found in the appendix 'ANACONDA2 built-in Spectrometers'.

To load the selected file, press 'Load file' after having selected the desired configurations. As files are being loaded into memory, they will be visible in the 'Files loaded'-list. The last loaded file's information will be shown in the file information window, which exists if it has been properly defined during data acquisition during an experiment. Selecting a loaded file results in that this file's information will be shown in the file information window.

To unload a file from memory, select a loaded file and press 'Unload file'.

In our example, we have loaded 4 different data files from 3 different experiments, stored at three different locations. The first three files have been loaded using local configurations whilst the last file was loaded with system configurations with the Laksman_TOF_e_XY spectrometer.

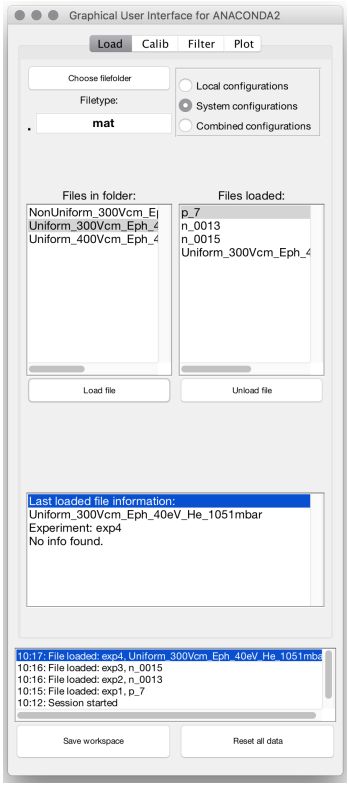

Figure 5: Files in selected folders with selected filetype can be seen.

3 Data treatment part 2: Data Calibration

[NOT YET CONSTRUCTED. Accessible via the ANACONDA2 kernel, and documented in ref. [1].]

4 Data treatment part 3: Data Filters

The goal of the 'Filter' tab (Figure 6) is to construct different conditions, that filters out all data that is not interesting when the experimental data is being plotted. A condition can also be referred to as requirements which a hit and/or an event has to fulfill in order to pass through to the plot. These requirements are defined by the user and are set such that only the data of interest will remain in the plot. A filter is either a single condition or multiple conditions, or a set of conditions, which is converted to a filter as data is being blocked by or passing through the condition(s).

The actions that can be done with data conditions is the following: Construction of new conditions, editing conditions, renaming conditions and removing conditions.[2]

By opening up the tree of data filters for the different experiments, selecting a condition results in that all its parameters will be shown.

Filters can be combined into a so-called 'Combined filter', which consists of one or many hits and/or events filters. It can also consist of other combined filters. The combined filter(s) then needs an operator that defines how the combined filters will work. The operators that can be selected from are 'AND', 'OR', 'XOR', 'HIT1' and 'HIT2'.

As an example, the 'AND'-operator states that all conditions that make up the combined condition have to be fulfilled, whilst the 'OR' operator states that at least one condition that is part of the combined condition has to be fulfilled in order to let the event and/or the hit through. The full specification of the condition operators can be found in the main documentation of ANACONDA2 [1].

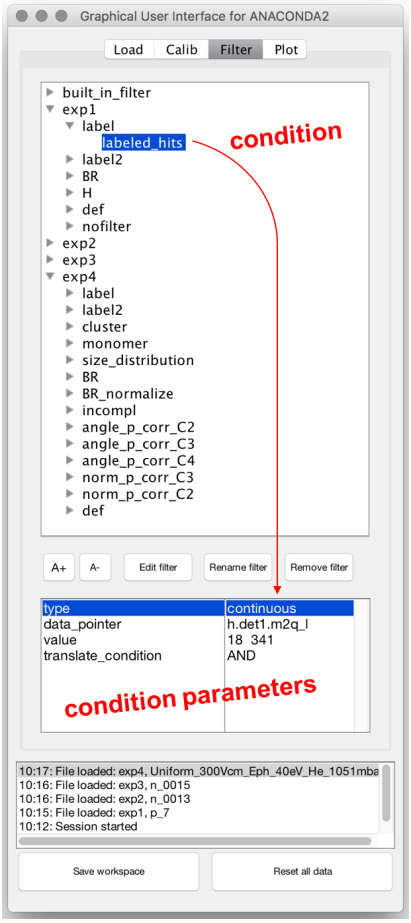

Figure 6: The filter tab.

[2]The condition tree fontsize can be increased and decreased by clicking on [A+] and [A-], resp.

If a combined condition is selected, the operator of the combined condition will be shown. If no operator has been defined, the standard condition parameter 'AND' will be shown as operator. To edit a field, double-click on the field to edit, which opens up a window with the previous value (Figure 7) that can be edited.

To construct a new condition, select one of the built-in conditions or the built-in combined condition, and drag-and-drop (DnD) it to the desired location. The parameters can then be edited. Conditions can also be drag-and-dropped between different experiments.

To rename a condition, press the 'Rename' button and type in the desired name. Note that the name has to start with a letter, and there cannot be any spaces or blanks - in these cases, use the underscore symbol.

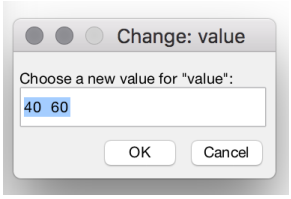

Figure 7: The 'Value' of the 'labeled_hits' condition is being edited.

To fully edit all conditions of a (single, not combined) condition, select a condition and press the edit button. This results in that the 'Edit condition'-window will open up for the selected condition (Figure 8), in which the condition value type can be selected, the values can be defined, translate condition can be defined or turned on or off, invert condition can be turned on or off, and data-pointer can be edited.

A data-pointer defines how the data is converted for the condition before the condition is applied onto the data[3], i.e. it points towards a real value (e.g. 'KER' (Kinetic Energy Release)). This is also known as a signal. Since the data-pointer 'points' towards a signal, the edit feature of the data pointer has been constructed such that the user can select signal that the data pointer points towards by using pre-defined signals in the 'Edit condition' window.[4]

[3]Note that this data-pointer is not linked to the plot type. E.g. a KER condition (i.e. a KER data-pointer) can be used while plotting e.q. TOF.

[4]More advanced users can have it completely editable as a text string by double-clicking the data-pointer field without opening the 'Edit condition' window, e.g. for a mathematical formula (e.g. "exp.h.det1.KER/3").

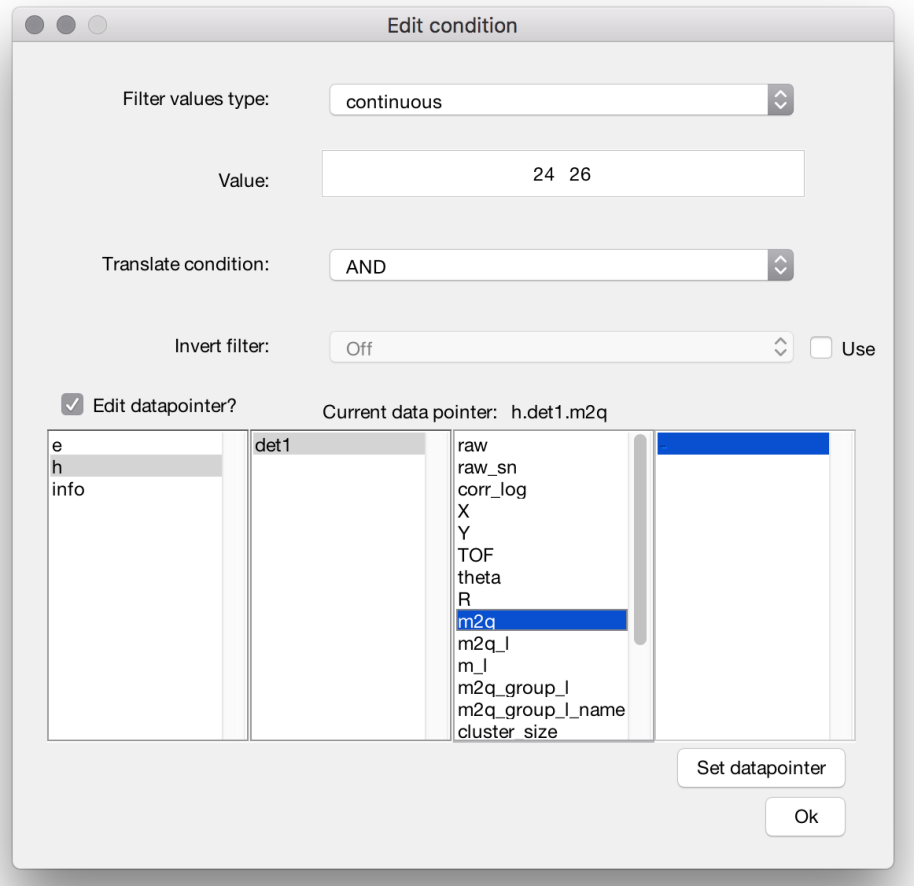

Figure 8: Edit all fields of a condition by opening up the 'Edit condition' window.

An example of a condition built upon one logical condition for e.g. a mass-spectrometry experiment is a 'hits-filter', see Figure 8. The data pointer in the edit condition window depicted in the figure is set as 'h.det1.m2q', which means that the condition is defined as a hits condition. The value type is set as continuous with the values 24 and 26, which means that hits which have a value between 24 and 26 will be allowed through. Since 'm2q' stands for 'mass-over-charge', i.e. not raw data, it has a unit in accordance with the calibration and conversion carried out previously - in this case, in unit Daltons. The translate condition is set as 'AND', which means that in order for an event to pass through, all its hits have to have values between 24 and 26. Hence, translate condition has to be defined for hits conditions, but not for events conditions.

Other translate conditions are:
- 'OR' - at least one hit has to have the specified value
- 'XOR' - only one hit exclusively has to have the specified value
- 'HIT1' - the first hit has to have the specified value
- 'HIT2' - the second hit has to have the specified value

Invert filter means that the condition will be inverted, i.e. that true becomes false and false becomes true (events or hits that would have passed through will be blocked, and events or hits that would have been blocked will pass through).

The filters are constructed as metadata for each experiment individually, based either on locally defined metadata and/or metadata pre-defined by the ANACONDA2's spectrometer settings (during the loading of the files). There are hard-coded built-in filters that are not connected to any experiment, which cannot be edited directly but can be added to different experiments and then edited. The filters associated with the various experiments are further on referred to as user-defined filters.

5 Data treatment part 4: Plotting

The goal of the 'Plot' tab is to prepare the experimental data for plotting by selecting pre-defined (common) plot configurations, or by constructing new plot configurations - either from pre-defined signals or by constructing new signals. We will now go through how to plot the data by using the GUI. The first step is always to select the experiments which are to be plotted by clicking in it/them (multi-selection is available by holding the shift-key whilst clicking on experiments). The plot tab has 3 'sub-tabs', as can be seen in Figure 9: Pre-defined plots, New plot conf(igurations), and New signal conf(igurations).

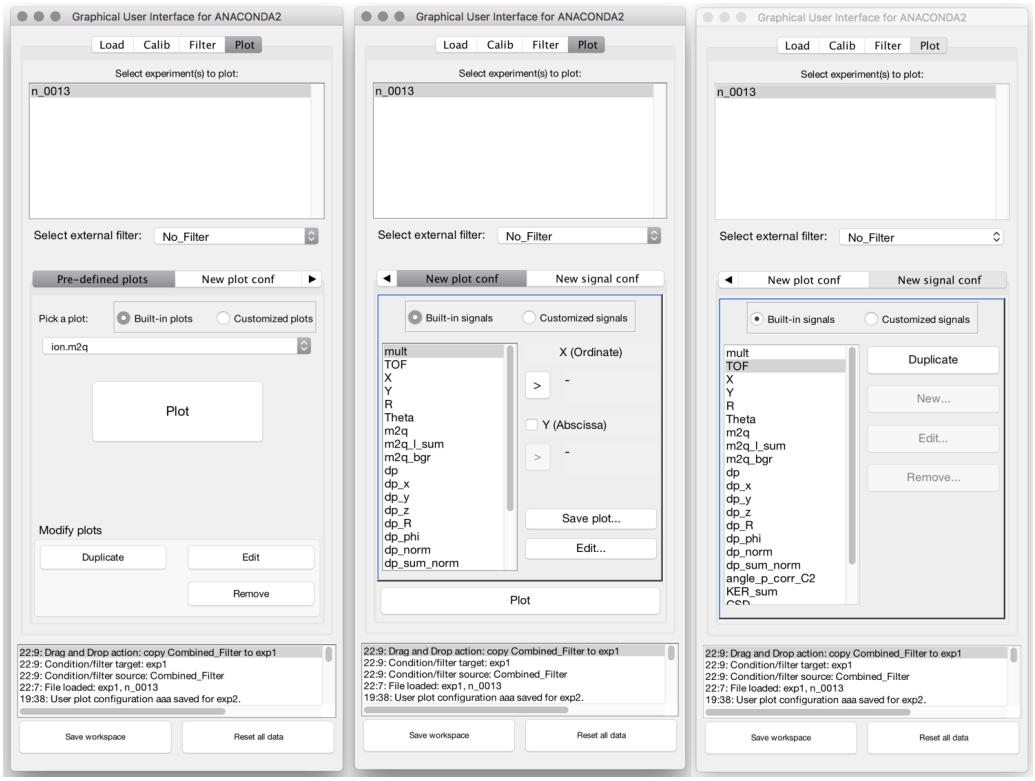

Figure 9: Plot tab's 3 sub-tabs: Pre-defined plots, New plot conf(igurations), and New signal conf(igurations).

In the first sub-tab, i.e. 'Pre-defined plots', the plot configurations are already set and ready to be plotted by selecting the desired plot configuration and hitting the 'Plot' button.

In order for data to be plotted, the signal(s) to be plotted and its (or their) associated metadata have to be defined, as well as their (if any) internal condition and if chosen, an external condition.

The fastest way to look at some results is by selecting a built-in plot, which are shown when the radio-button is set on 'built-in plots'. One can select to use an external condition or not, and then simply plot the data by pressing the plot button (left window in Figure 9).

The 'pre-defined plots' sub-tab gives access to two different lists: Built-in and customized plot configurations. The built-in plots are hard-coded and loaded as standard plot types with the experiment during load, whilst the customized plots are the plots that have been constructed by the user in 'New plot conf' sub-tab. The 'New plot conf' and the 'New signal conf' sub-tabs also give access to two different lists: Built-in and customized signals. New signals can be constructed whilst customized signals can be edited and removed in the 'New signal conf' sub-tab, and new plot configurations can be constructed in the 'New plot conf' sub-tab. We will now walk through the whole process, from constructing a new signal and using the new signal to construct a new plot configuration, followed by plotting this and see if we need to do some further editing of the plot configuration.

5.1 Constructing a new signal

A new signal can be constructed in two ways: Selecting a previously existing signal (built-in or customized) and pressing duplicate. If the previously existing signal that is to be duplicated was customized, an input dialogue will prompt the user for a new name on the duplicate of the customized signal. If the previously existing signal was built-in, it will be copied to customized signal. The other way to construct a new signal is to select the radiobutton 'Customized signals' and then pressing the button 'New...'. We will go through an example of how to construct a new signal by first duplicating and then editing the duplicated signal.

Open the 'New signal conf' tab and ensure that the built-in signals' radio-button is selected, since so far we have not constructed any new signals, i.e. the customized signals list is empty. Then select a signal, which in our case is 'TOF' (time-of-flight). We will, for the construction of the new signal example, refer to Figure 10's different numbering, in which each action has been numbered as a sequence for the construction of our new signal.

When the signal (in our case, 'TOF') has been selected, press 'Duplicate' (1). This copies the selected (TOF-) signal to the user-defined signals, which can be found under 'Customized signals'. Select the 'Customized signals' radio-button (2), select the duplicated signal ('TOF') in the signals list and press the 'Edit...' button. This opens the dialog window 'edit signal'. This window can also be reached by pressing the 'New...' button, with the difference that all the value of the fields 'Name', 'Range', 'Binsize', 'Data pointer' and 'Axis label' are empty.

We begin by editing the name, bin-size and axis label (4) to the values shown in Figure 10, marked by an arrow and a '4' above (bottom-left corner window). We then want to look at m2q (mass over charge) rather than TOF. We can switch by choosing a new data pointer that points towards m2q, which we can either do by editing directly in the data pointer's value field or by using the 'Data pointer selector', of which we will do the latter.

Since m2q is a hits property, we select 'h' in Listbox 1 (5), which then shows the possible subfields in Listbox 2. In our case, 'h' has only one subfield, i.e. only one detector 'det1', and which we thus select (6). This opens up all subfields that can be used for our detector in Listbox 3, where we see and select 'm2q' (7). Since 'm2q' has no subfields, nothing can be shown in Listbox 4 (which hence remains empty). This enables the 'Set datapointer' button, which we press (8). This changes the value of the field 'Data pointer'.

Note that if the selected field in listbox 3 would have a subfield, it would show up in listbox 4. If the selected field in listbox 4 would also have a subfield, all lists would be moved one step to the left. Selecting any field in listbox 1 then resets the data pointer selector.

We finish by pressing the 'Ok' button (9) and see that the name has changed in the list of customized signals (since we chose to edit its name).

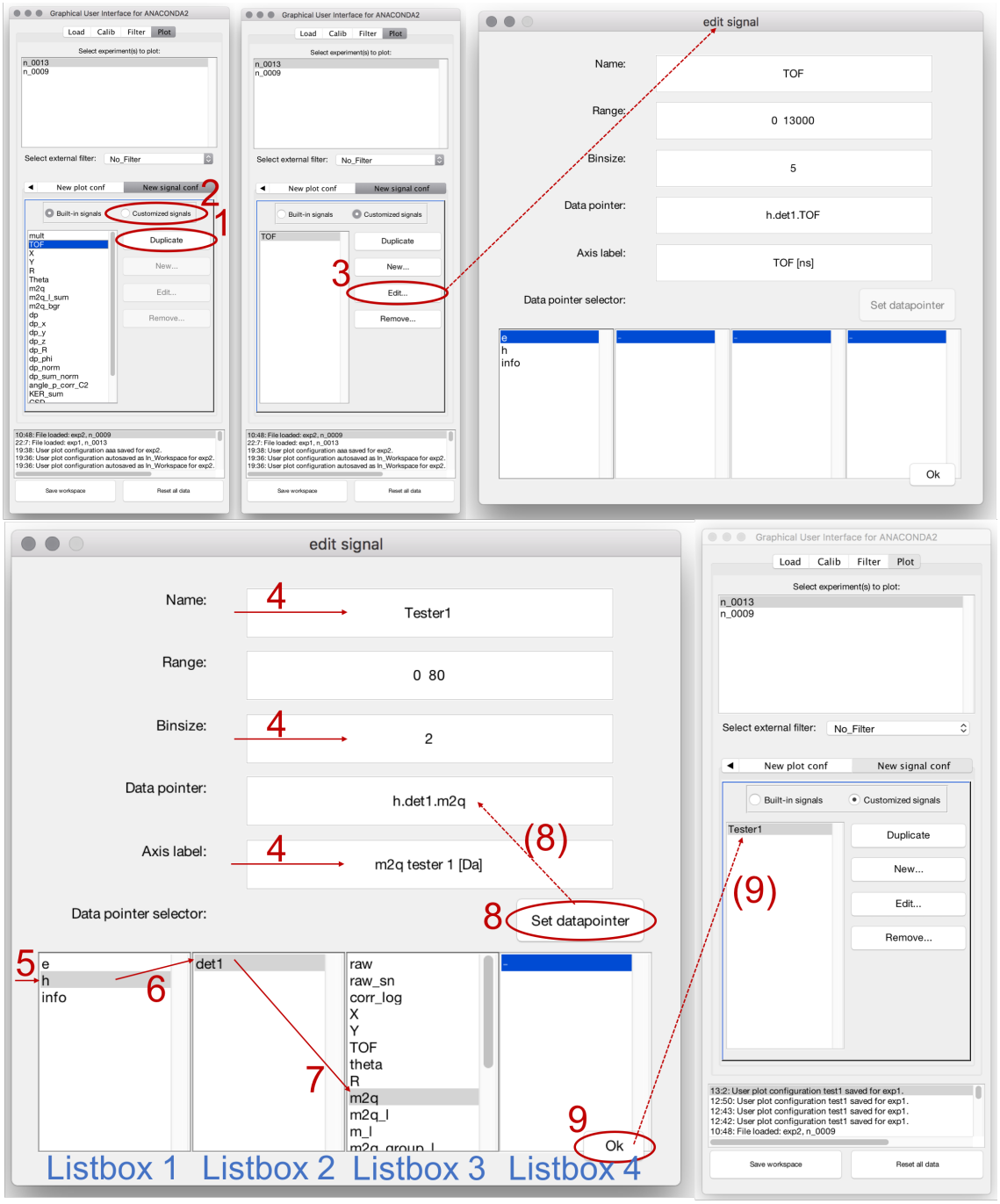

Figure 10: Constructing a new signal by duplication followed by editing.

5.2 Constructing a new plot configuration

In order to construct a new plot configuration, open the 'New plot conf' sub-tab, select a signal and put it as Ordinate ('X') and/or Abscissa ('Y'). There are many possible combinations between different signals in order to construct a plot configuration out of them. Note that in order to have intensity as abscissa, the check-box for 'Y (Abscissa)' has to be un-checked, and to use something else than intensity as abscissa, click the checkbox so that it is checked. As an example, we will create a new plot configuration with our new signal ('Tester1') as ordinate and a built-in signal ('Theta') as abscissa.

We want to use the user-defined (customized) signal ('Tester1') as ordinate. Hence, ensure that the built-in signals' radio-button is selected. Once selected, the user-defined signals will show in the signals listbox. We will, for the construction of the new plot configuration example, refer to Figure 11's different numbering, in which each action has been numbered as a sequence for the new plot conf. construction.

When the customized signal for ordinate ('Tester1') has been selected, press the set signal [>] (1 in Figure 11) to set it as ordinate. Then click to check the abscissa checkbox (2), and open the list of built-in signals (3). Select the signal to be used as abscissa ('Theta') and press the set signal (4) to set it as abscissa. Save the plot configuration (5), write the name of the new plot configuration (6) - which we call 'Testplot1' - and press the 'Ok' button (7). Then switch over to the Pre-defined plots (8) and ensure that the radio-button 'customized plots' (9) is selected. Then open the popup-box with the customized plots (10) and select the constructed plot configuration ('ion.Testplot1'). Note that the plot configurations are named as detector-type ('ion'), followed by a separator (a dot) and plot configuration name ('Testplot'). In order to plot with the saved and customized plot configuration, press the plot button (11). In our case, it generated Figure 12. The plot seems to be empty due to bad signal values when we constructed our signal ('Tester1'). We will walk through the process of editing a plot configuration in the variable editor in section 5.3.

The plot configuration can also be used for plotting the experiment without saving. One way is to press the plot button directly (X in Figure 11) in the 'New plot conf' sub-tab (instead of in the pre-defined plots' sub-tab). The other way is to select the 'In_Workspace' plot configuration in the popup-box with the customized plots (10) in the pre-defined plots' sub-tab, which is also defined by its detector specie ('ion'). The saved (customized) plot configurations can be edited and removed by first being selected (10) and then pressing the edit/remove button.

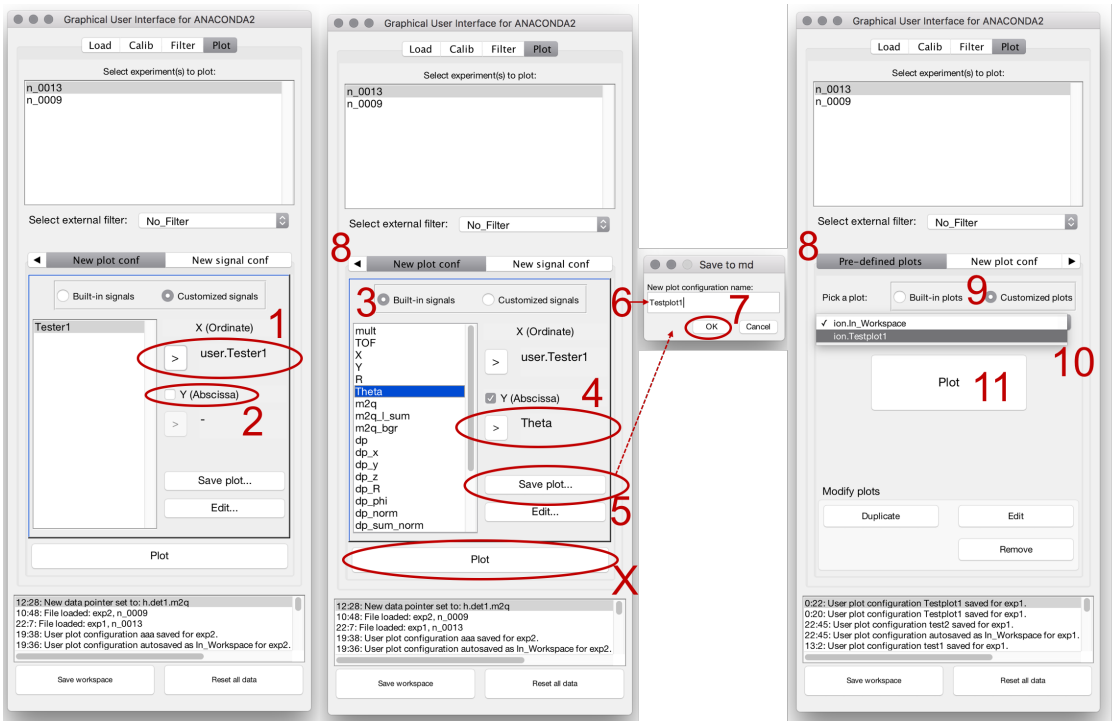

Figure 11: Constructing a new plot configuration from the previously constructed (customized) signal 'Tester1' as ordinate and the built-in signal 'Theta' as abscissa.

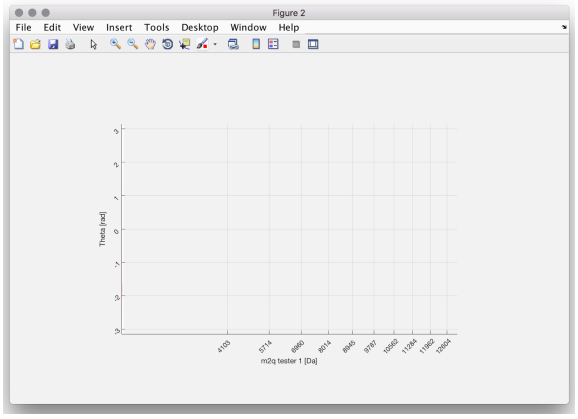

Figure 12: The resulting plot from the process described above, with the customized 'Tester1' signal on the x-axis and the built-in 'Theta' signal on the y-axis.

5.3 Editing with the variable editor

As can be seen in Figure 12, the plot seems to be empty. This is probably due to that something is 'bad' with at least one of the signals used as the plot configuration, of which our signal ('Tester1) is probably the cause. Hence, we will look into it with the variable editor.

In order to open the variable editor, we select the customized plot ('Testplot1') in the popup-box with the customized plots in the sub-tab 'Pre-defined plots' and then press the 'Edit' button. This opens up the variable editor in the main MATLAB window ('Variables - md_GUI.mdata_n.exp#.plot.user.det1.Testplot1'), as can be seen in Figure 13. In the variable editor, we see 4 structures: Figure (which contains information about the plot container, such as color, position and size), axes (which contains information about the x- and y-axis, such as limits, ticks, tick-labels, labels, etc.), hist (histogram, which contains information about the plot data, such as bin-size, range, number of dimensions and data-pointers) and GraphObj (graphical object properties, such as image type, marker type, marker edge color, etc.).

Since we cannot see anything in Figure 12, we make an initial guess that this problem is caused by the axes. We hence open up the axes structure by double-clicking it, resulting in that a new tab is opened up in the variable editor window (Figure 14). We investigate the XTicks by double-clicking it and find values from 4103 to 17791. This would mean very large masses, but these values come from that we previously duplicated a time-of-flight signal with time units instead of mass-over-charge units. Hence, we change those to values we know will show up for the loaded experiment, which happens to be carried out on pyridine and pyridine fragments. We also change the XTickLabel to show the name of the fragments they are associated with, as described in the Table 5.3 below, and the XLim values from 0 to 80.

Table 5.3: X ticks and X tick labels, i.e. molecules and their mass-over-charge.

XTickLabel	*XTickvalue*
H^+	1
C^+	12
C_2^+	24
C_3^+	36
C_4^+	48
$C_5H_5N^+$	79

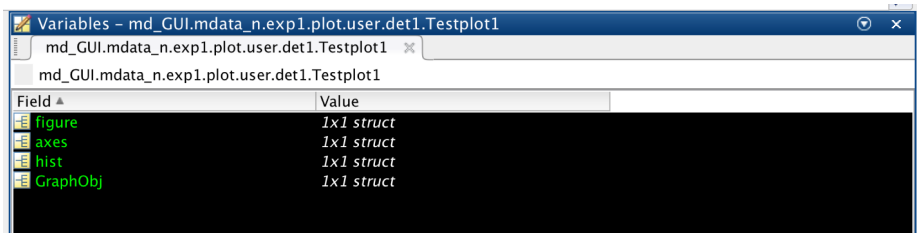

Figure 13: Variable editor for the Testplot1 .

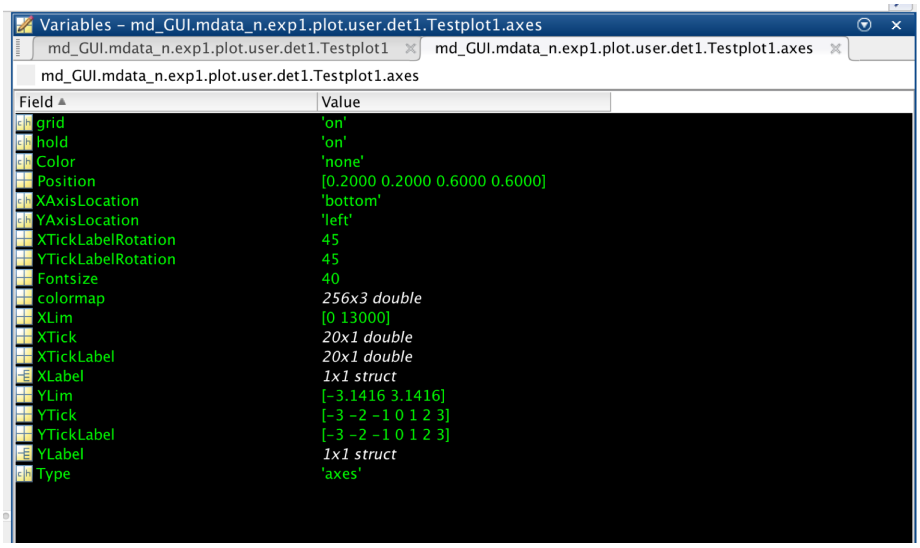

Figure 14: Variable editor for Testplot1's axes properties.

When we plot this plot configuration, the result is Figure 15. We notice that the bin-sizes seem to be too large in the x-direction, so we edit the close the variable editor for the axes and open the variable editor for the histograms by double-clicking on the hist structure, and then on the bin-size. We change the value of the first column from 2 to 0.5, and then we plot the plot configuration again. This results in Figure 16 when we plot this plot configuration.

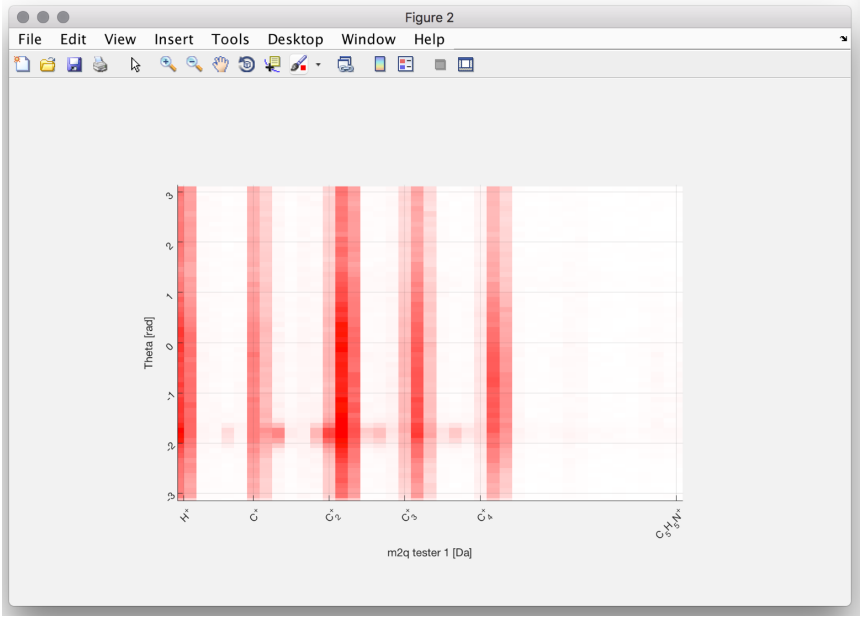

Figure 15: Our signal in units mass over charge plotted against theta.

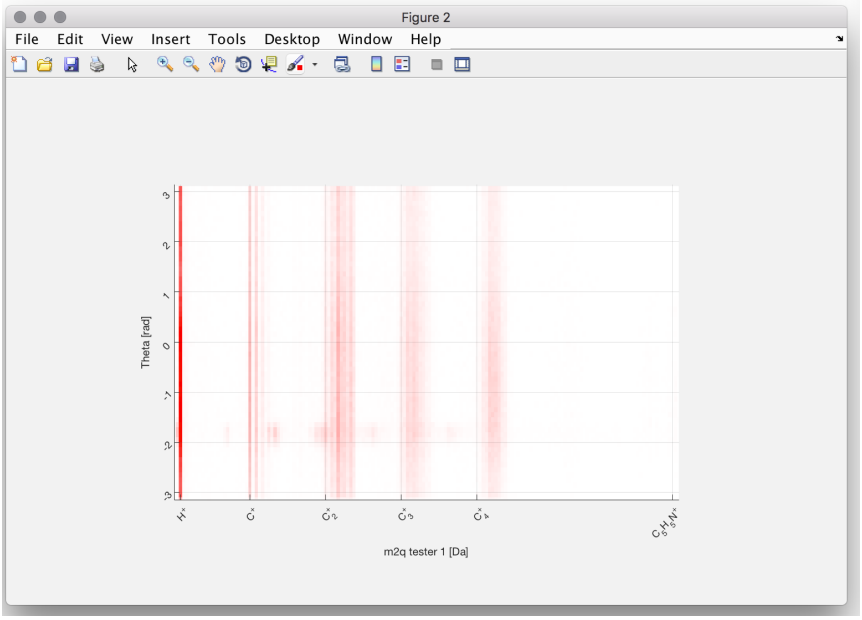

Figure 16: Our signal in units mass over charge plotted against theta after correcting the bin-size of our signal from bin-size 2 to bin-size 0.5.

5.4 Plot data selection example: Pyridine

In order to demonstrate how data can be selected by using various data conditions, the same experiment's PEPIPICO (photo-excitation photo-ionization photo-ionization coincidence) map (m2q for HIT1 and HIT2) will be plotted three times; once without any filters applied and twice with filters applied. Note that here we used the term filter instead of filter, since we refer to the data after a condition was applied to the data, i.e. filtering.

First we use no filter to look at the entire map in Figure 18, whilst the second time we want to look at only 2 specific events (in the same plot): A kinematically incomplete detection (No. 1) and a(n) (almost) kinematically complete detection (No. 2). The third time we look at only one event, which is a kinematically complete detection (No. 3).

The experiment we analyze was carried out at MAXLAB in 2011; pyridine studied with synchrotron radiation. In the analyzed experimental data, the energy of the photons was 399.1eV.

No. 1: For the kinematically incomplete detection we want to observe the channels corresponding to N^+ (HIT1, m2q = 14) with C_3H^+ (HIT2, m2q = 37), which together corresponds to a detection of $C_3H^{++}N$ and hence, a total loss of 2C and 4H. For this, we construct a combined condition with a hits condition: $13.5 < m2q < 14.5$, and an events condition: $50.5 < m2q < 51.5$. This combined condition will be named 'FiltNo1' and have an 'AND' operator, see Figure 18E.

No. 2: For the (almost) kinematically complete detection we want to observe the channels corresponding to CN^+ (HIT1, m2q = 26) with C_4H^+ (HIT2, m2q = 49), which together corresponds to a detection of $C_5H^{++}N$ and hence, a total loss of 4H. For this, we construct a combined condition with a hits condition: $25.5 < m2q < 26.5$, and an events condition: $74.5 < m2q < 75.5$. This combined condition will be named 'FiltNo2' and have an 'AND' operator, see Figure 18F.

No. 3: For the kinematically complete detection we want to observe the channels corresponding to CHN^+ (HIT1, m2q = 27) with $C_4H_4^+$ (HIT2, m2q = 52), which together corresponds to a detection of $C_5H_5^{++}N$ and hence, no loss. For this, we construct a combined condition with a hits condition: $26.5 < m2q < 27.5$, and an events condition: $78.5 < m2q < 79.5$. This combined condition will be named 'FiltNo3' and have an 'AND' operator, see Figure 18B,C.

For plotting No. 1 and No. 2 together in a single plot, we have to combine their condition conditions, which we name 'FiltNo12'. This means that we combine two combined conditions with an 'OR' operator, see Figure 18D and Figure 17.

Figure 17: A two-level combined condition: Filt12 consists of the two combined conditions FiltNo1 and FiltNo2, of which each consists of a hits-condition and an events-condition. The translate condition operator between FiltNo1 and FiltNo2 is 'OR', i.e data is allowed through if either of these are fulfilled. The hits and events conditions have the 'AND' translate condition operator between them, meaning that both conditions have to be fulfilled in order to let data through.

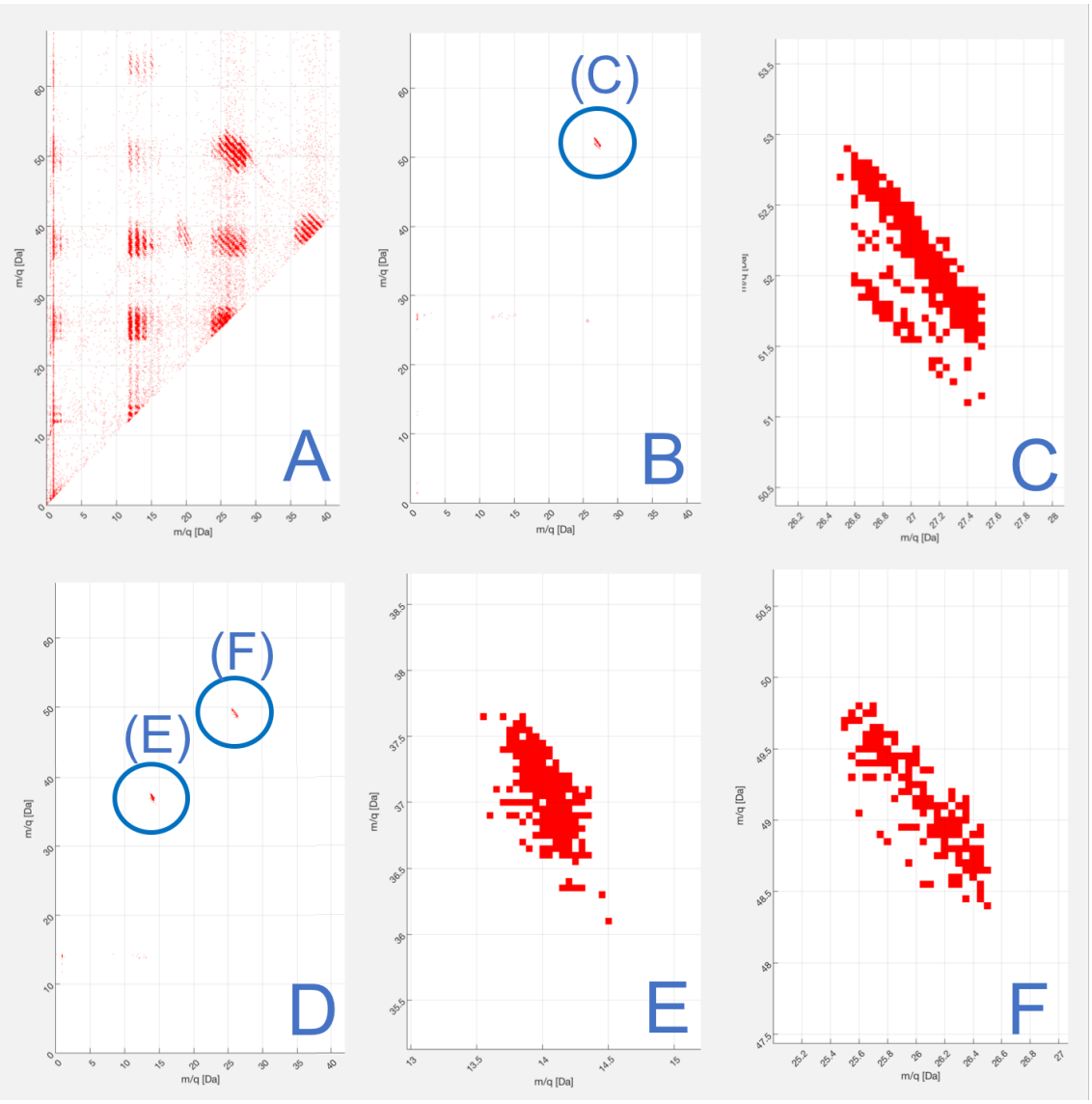

Figure 18: A PEPIPICO plot for an ion-ion coincidence experiment with pyridine.

5.5 Analysis of the plot data selection example: Pyridine

The PEPIPICO ion-ion coincidence experiment with pyridine has been plotted with various filters in Figure 18. In subfigure A, no filters has been applied, i.e. that the full PEPIPICO map has been plotted (all collected double-coincidence events). Each tilted red bar, seen in and enhanced in subfigures C, E and F, represents different event species.

Since the experiment was carried out on pyridine with a photon energy of 399.1eV, which corresponds to the Nitrogen's 1s atomic edge (meaning that a 1s electron of the Nitrogen atom has a high probability of absorbing the photon and being kicked out as a photo-electron), the probability is high for an Auger electron to be emitted as a consequence of the ionization of a core-level electron. The photo-emission of a N1s-electron followed by the emission of a N2p Auger electron has been illustrated in Figure 19.

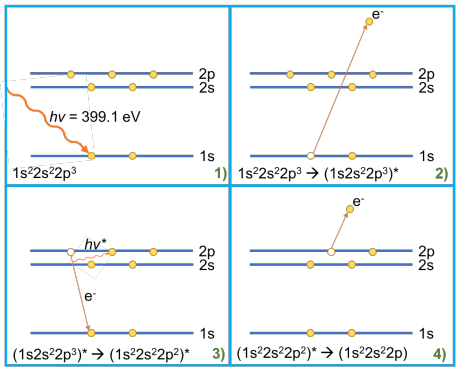

Figure 19: Core-level ionization of a Nitrogen atom: Photo-emission of a N1s (photo)electron followed by an Auger electron. The hv* photon has not been observed experimentally, and is hence referred to as an imaginary photon. The atomic configuration (and transition) is located at each box's bottom-left corner.

In subfigure B of Figure 18, enhanced in subfigure C, we see an event consisting of one hit with m2q = 27 Da, and a second hit with m2q = 52 Da. The total m2q becomes

$$[m2q]_{event} = [m2q]_{hit1} + [m2q]_{hit2} = 27 + 52Da = 79Da.$$

The mass-over-charge for a singly charged pyridine molecule is $m2q(\text{pyridine}^+) = 79$ Da. This means that the total undetected mass (undett) following this event is

$$[m2q]_{undet,C} = [m2q]_{tot} - [m2q]_{event} = 79 - 79Da = 0.$$

Hence, we see that all fragments have been collected for this event, which is referred to as being kinematically complete. A possible dissociation process is sketched in Figure 20 for process X.

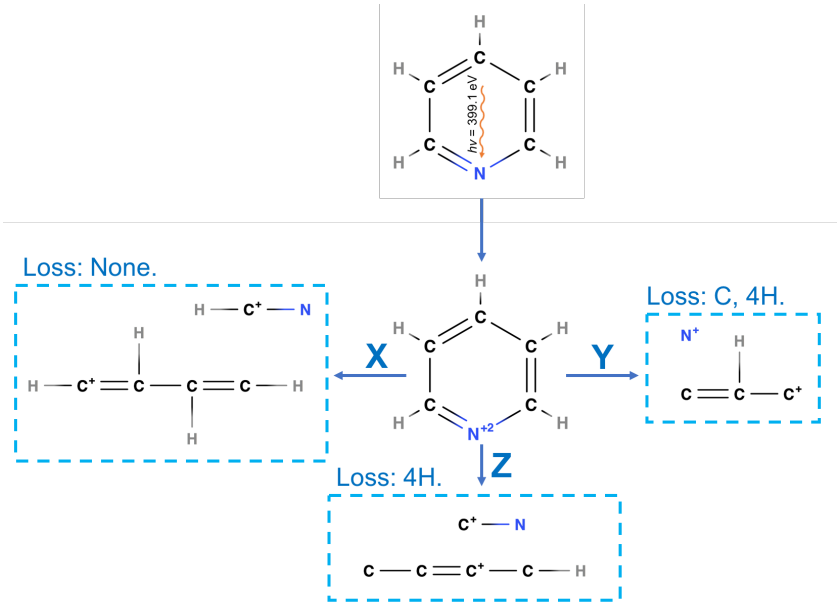

Figure 20: Dissociation of a pyridine molecule by double-ionizing the Nitrogen atom. Three detected events have been depicted: X, Y and Z, where X is the only event for which all fragments have been collected by the detector. In the figure, loss refers to the atoms that were not collected by the detector.

In subfigure D of Figure 18, enhanced in subfigure E and F, we see two events. One event (subfigure E) consists of a hit with m2q = 14 and another hit with m2q = 37, resulting in that the total m2q becomes 51. The other event (subfigure F) consists of a hit with m2q = 26 Da and another hit with m2q = 49 Da, resulting in that the total m2q becomes 75. Thus, the total undetected mass for the events E and F are

$$[m2q]_{undet,E} = [m2q]_{tot,E} - [m2q]_{event,E} = 79 - 51Da = 26Da,$$

$$[m2q]_{undet,F} = [m2q]_{tot,F} - [m2q]_{event,F} = 79 - 75Da = 4Da.$$

We can thus see that all fragments have not been collected for either of these events, which are hence referred to as being kinematically incomplete. Possible dissociation processes are sketched in Figure 20 for process Y and Figure 20 for process Z, for E and F, resp.

5.6 Analysis of the kinetic energy release (KER)

Plot the Dalitz plots for pyridine 399.1eV, Helium 300Vcm 40eV and Helium 400Vcm 43.5eV and talk about them a bit. What can we see and conclude? Make it short.

References

[1] Bart Oostenrijk, *Description of the Coincidence data analysis software*, Department of Physics at Lund University, 2nd edition, 2018.

[2] Joakim Laksman, *Nuclear motion in molecular ions studied with synchrotron radiation and multicoincidence momentum imaging spectrometry*, Department of Physics at Lund University, PhD thesis, 2012.

[3] Adrian Wuthrich, *Dalitz Plots and Hadron Spectroscopy*, Faculty of Science, University of Bern, MSc thesis, 2005.

[4] Aaron Woods Harrison, *Investigations of Two- and Three-Body Molecular Photodissociation Dynamics by Fast Beam Photofragment Translational Spectroscopy*, Graduate Division (chemistry), University of California, Berkeley, PhD thesis, 2014.

[5] Benjamin Bolling, *Studies of pyridine in water clusters with synchrotron radiation and 3D momentum spectrometry*, Department of Physics at Lund University, BSc thesis, 2017.

Appendix 1: Data process schematics

The data process schematics for plotting has been illustrated in Figure 21, when the plotting has started, the GUI first checks if a condition has been selected. If no, the internal conditions (if any) will be used, and if yes, the selected condition will be used and then (in this version) overwrites the built-in filter. This becomes part of the experiment MetaData, which together with the selected signal(s) are plotted by the software in the way it was defined by the user in the plot configurations.

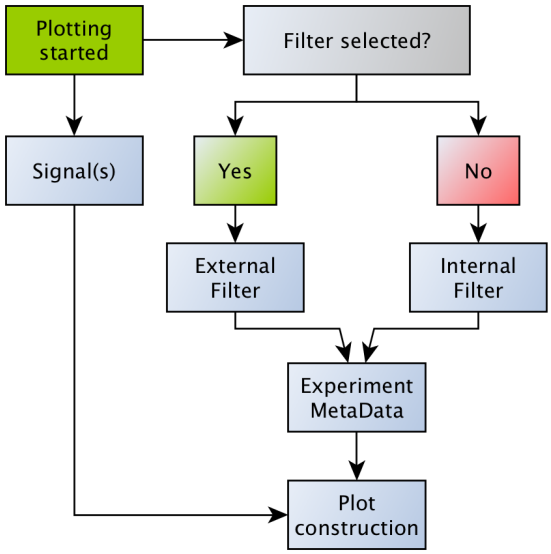

Figure 21: Plot process data flow schematics.

Appendix 2: *ANACONDA2* built-in spectrometers

EPICEA is a spectrometer at SOLEIL.
Laksman_TOF_e_XY, will be Oostenrijk_2D in the future.
Laksman_TOF, will be Laksman_3D in the future.