Description of the Coincidence data analysis software

Monday 23$^{\text{rd}}$ May, 2022

Contents

1

1 Introduction

This document presents the contents of the data analysis package 'ANACONDA 2', or 'ANACONDA', which allows the analysis of data recorded in experiments that use single-particle detectors (e.g. ions, electrons, photons ...), for example to study the correlation between these particles. These are often called 'coincidence' spectroscopic treatment methods. This document presents three parts of the software. The structure of the event-based data, the structure of the metadata and finally the package containing all functions.

2 Data structure

The data is stored in a matrix of so-called *hits*, which is a single count/detection registered on a detector. The acquisition software can group these hits, because they are expected to be related to each other; this is called an *event*. The start of an event is usually defined by a trigger (which could be a hit on one of the detectors), the duration of an event can be defined by a timeframe from the event start.
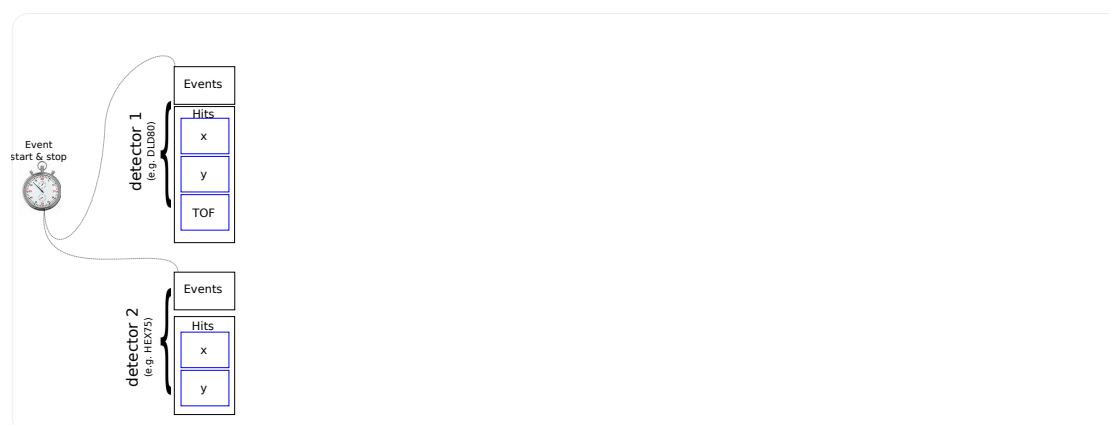
Figure 1: The synchronization of events by a common 'clock'.

For example, an event from an experiment on H_2O is the registration of two hits on the ion detector: one from H^+ and after that one from OH^+, and the registration of two hits on the electron detector. Note that the number of hits is generally different for different detectors.

The matrix of hits only contains the data that is given along with every hit (for example X, Y and T), but it does not show to which event the hit belonged. The grouping of events is done in another matrix, that contains the *event pointers*. The list of event pointers contains the index of the *first hit* of that event. This

structure is chosen, because it can easily be extended when the measurement uses multiple detectors. Important to notice is that the events from different detectors are synchronized; every event start is measured at the same instant in time.

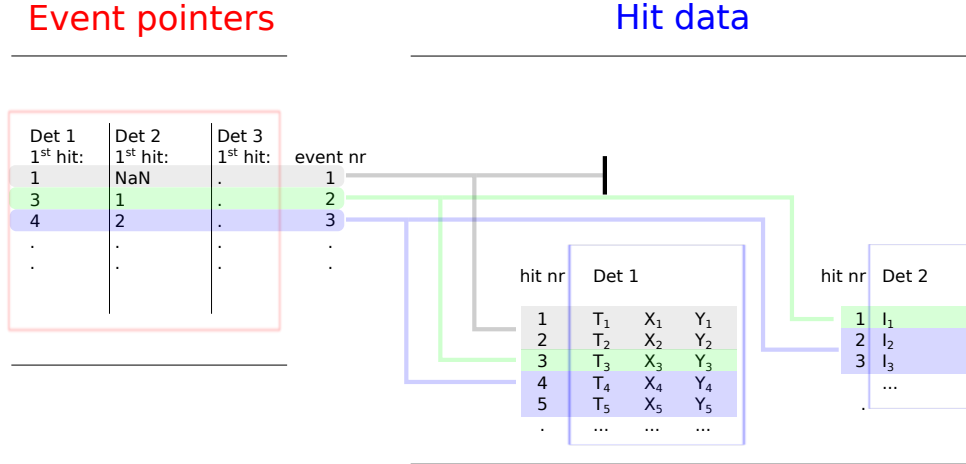A schematic of the data structure is shown in Figure 2.

Figure 2: Schematic of the data structure used in the software. The 'event pointers' array on the left stores the indeces of the first hit of the event, for each different detector used (in this case two detectors). On the right, the hit arrays are drawn, with the hits coming from these detectors. Note that each detector can give another number of hits, and a different number of signals (e.g. X, Y, T, Intensity). In this example, there are no hits recorded on detector 2 during event 1, which is denoted by 'NaN' (not-a-number).

The naming in the .MAT file is such, that all data is the subfields 'e' (for events) and 'h' (for hits). An example of a set of raw files is given below:

```
exp1.e.raw ;%  event pointers. size: [nof_events, nof_det].
exp1.h.det1.raw ;% detector 1 hits [nof_hits(1), nof_signals(1)]
exp1.h.det1.raw_sn ;% cell with signal names from det 1 [1,
    nof_signals(1)]
exp1.h.det2.raw ;% detector 2 hits [nof_hits(2), nof_signals(2)]
exp1.h.det2.raw_sn ;% cell with signal names from det 2 [1,
    nof_signals(2)]
```

The name 'exp1' can be changed, so that several different experiments can be opened at the same time.

3   Histogram data

In an increasing number of experiments, the data is not anymore collected in a hit-to-hit fashion, but rather as shot-to-shot histograms. For instance, in the case of short and intense laser or (X-)FEL pulses. Rather than individual pulses with e.g. a position and time of impact, the incoming data consists of histograms of *all* hits masses, X-positions, and Y-positions in an event.

The data treatment could then for instance involve only the visualization of those histograms, or a covariance treatment. The data will be stored in a different way.

TODO

3.1   Normalization

In case of histogram data, the amount of counts one often depends on external factors, for example the measurement time, the FEL power of each shot, the amount of ions present in the trap, their spatial distribution, and possibly fluctuations in the ESI source. Therefore, this signal needs to be normalized before one can deduct conclusions on the system under study.

Total ion count   When normalizing on the total measured counts, namely all fragments and the ion parent, the relative intensity might be interpreted as a relative yield. This type of normalization makes the signal still dependent on:

- The overlap of FEL with the ions in the trapped volume

- The FEL light intensity

Note that both these aspects can fluctuate over time and thereby influence the normalized ion yield.

Normalization of mass spectra

4   Metadata

The metadata contains all the information *about* the data, such as conversion parameters, but also filter and plot parameters. The package comes shipped with 'default' parameters, which can be used when the data is first imported by the user. It is recommended to copy the default parameters close to the local files, such that they are not changed, for instance when the package is upgraded. The default metadata in the package is defined for each spectrometer, since each spectrometer

Figure 3: The Normalization schemes to choose from. 'Fr' is an abbreviation of 'Fragment', 'Int' stands for 'Integrated', 'Wvnr' stands for 'Wavenumber', 'En' stands for (photon) 'Energy' (or wavenumber)

has different numbers of detectors, with different measurement signals (position, time, etc . . .).
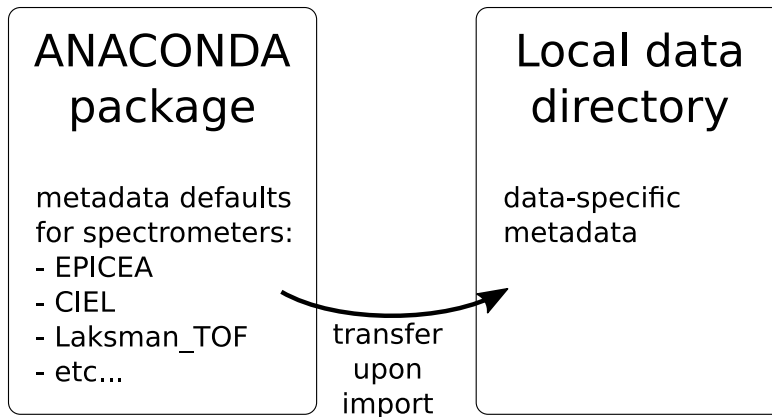

Figure 4: Default metadata (or configuration data) are defined in the package, to be copied upon first import of a new data file.

4.1   Metadata structure

The metadata contains all information needed to correct, convert, fit and visualize the data.

The metadata structure is divided into different categories:

sample , e.g. atomic mass, expected fragment masses, constituent masses.

photon beam , e.g. the photon energy, intensity, duration, etc

spectrometer , e.g. the name, voltages and relevant dimensions of the used spectrometer are listed here.

detectors , e.g. the names and properties of the detectors are stored in here.

correct The parameters needed to execute corrections onto the raw data, before conversion. For example, translation in X and Y to move the centre of detection into the origin of the coordinate system.

calibrate The information needed to perform the calibrations. Note that these are not the actual calibration factors, they are stored in the 'convert' field.

fit The fitting parameters.

convert The conversion factors (sorted in terms of detectors), such as mass to charge conversion.

7

plot The user-preferred plotstyle.

Most fields in the metadata are obvious to understand. We elaborate on a few that might be cause of confusion.

sample   We elaborate on the definition difference between 'constituents' and 'fragments' here:

- Constituents: The building block of which the sample consists. Example: a water-ammonia mixed cluster has consituents water and ammonia

- Fragments: The expected fragments from the sample. Example: a water-ammonia mixed cluster has the expected fragments of hydrogenated water-ammonia mixed clusters.

4.2   Storage

The metadata is stored as a struct, with the above categories as their fieldnames.

The metadata, or data settings, belong to a separate datafile. They are stored in a separate file with the same base name as the main datafile (*'filename.mat'*), but with the addition *'md_.m'* before the filename, so *'md_filename.m'*. This is done to make it easy to manually copy metadata files to different datafiles. For example: reading the datafile *H2O_003.mat* will read its metadata from the file *md_H2O003.m*. The metadata is stored in a plain-text m-file, which makes it possible to read and change parameters from outside MATLAB.

```
exp1_md.sample
exp1_md.photon
exp1_md.spec
exp1_md.corr
exp1_md.calib
exp1_md.fit
exp1_md.conv
exp1_md.plot
```

5   Data flow

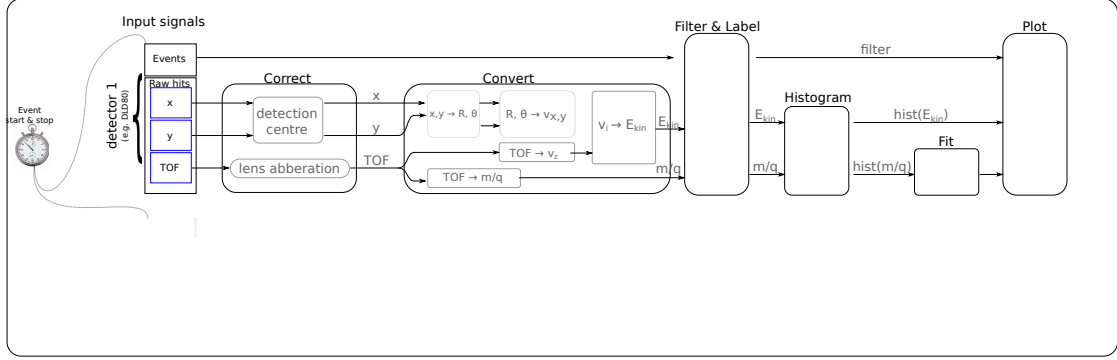In this section, the transition from raw hits to a calibrated and physically interpretable signal is described.

Figure 5: Example of data flow

6 The package

The package is defined as the collection of functions (or 'methods') that are part of this software. The methods are divided up in several categories:

import/output (IO) : Data Input/Output functions

correct : Correction of the data, for example abberrations.

convert : Conversion of the corrected data to other variables (signals)

filter : The formation of filters that view only parts of the data

plot : Show histograms of many kinds, and other graphics tools

calibrate : Methods to verify the conversion

macros : Meta-functions that automize the treatment and analysis of (multiple) experiments.

fit : Methods to characterize histograms from models.

theory : Methods to visualize or use (statistical) models.

general : General MATLAB functions used in different places in the package.

6.1 Input Output (IO)

Importing DLT files The importing of data from DLT files is done with software written by Erik Mansson.
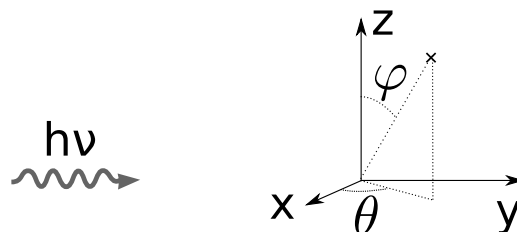
Importing KoboldPC (LMF) files Not yet implemented.

handling of multiple files Not yet implemented.

Combining two files Not yet implemented.

Comparing two files Not yet implemented.

Figure 6: Definition of coordinate system used in this Software. The x-axis aligns with the polarization and the molecular beam, the Y-direction aligns with the photon beam, and Z aligns with the detector axis.

6.2 correct

This section lists all the corrections that can be performed. It is logged in the datafile whether a certain correction has already been performed, in the *exp_name.h.det_name.corr_log* field. First of all, the coordanite system used in this software is defined in Figure 6.

6.2.1 detection image translation

Metadata parameters used by the detector image translation: .

```
exp_md.corr.det1.ifdo.dXdY      = true; % Does this data need detector
    image translation correction?
exp_md.corr.det1.dX           = -0.5; %[mm] distance the center of
    detection is displaced left of the origin of the raw image;
exp_md.corr.det1.dY           = 3; %[mm] distance the center of
    detection is displaced up the origin of the raw image;
```
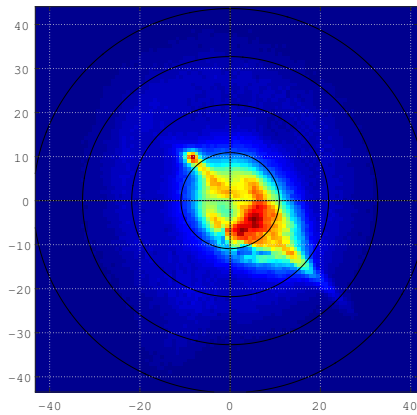
Figure 7: Detector image of all hits before centring
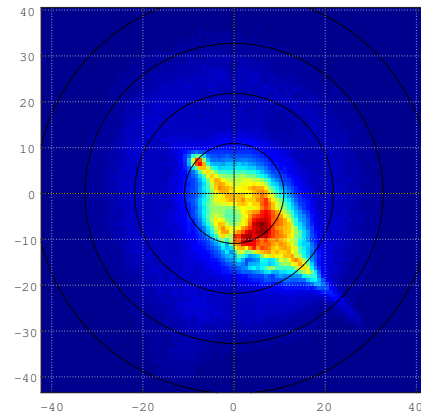

Figure 8: Detector image of all hits after centring. Final values: dX = -0.5 mm, dY = 3mm

6.2.2 detector image rotation

As seen in Figure 7, the detector image shows a line of higher intensity towards the left lower corner of the detector. This is believed to originate from heavier clusters from the molecular beam. The detector image is rotated such, that this line is along the x-axis.

Metadata parameters used by the detector image rotation: .

```
exp_md.corr.det1.ifdo.dTOF    = true; % Does this data need detector
   absolute TOF correction?
exp_md.corr.det1.dTheta       = 44;  %[deg] rotation of hits around the
   raw image centre (anticlockwise);
```

6.2.3 Absolute TOF translation

The recorded time of flight can have a deviation from the actual one, due to timing differences. These can for instance originate from different signal propagation times of the trigger and the detector signal.
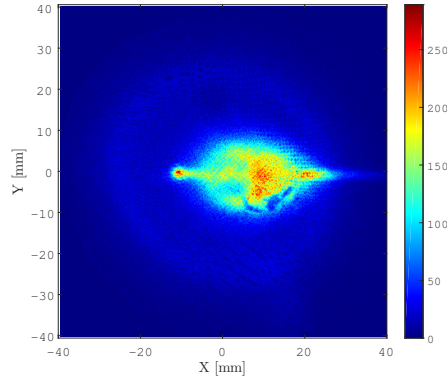
Figure 9: The detector image rotated such that the molecular beam is aligned with the x-axis. In this case, it means an anti-clockwise rotation of 44 degrees.

Metadata parameters used by the absolute TOF translation: .

```
exp_md.corr.det1.ifdo.dTheta   = true; % Does this data need detector
    image rotation correction?
exp_md.corr.det1.dTOF          = -16.7; % [ns] The difference between
    signal propagation times of trigger and hits
```

6.2.4 TOF deviation due to detector-drift tube voltage mismatch

If the MCP front voltage is not kept at the same voltage as the drift tube, the local field in front of the detector will not be uniform, but curved instead (see Figure 10. This will cause a distortion in the actual TOF, compared to the one predicted assuming a pure drift of the ions in the drift tube. The Time Of Flight will decrease if the detector front is kept at a higher absolute voltage. Moreover, this influence is dependent on the position on the detector. If the ion approaches the detector out of center, the deceleration is later, because the field inhomogenity starts later. Therefore, a correction dependent on radius is needed. First, the correction in TOF is presented, and consequently the correction in splat radius is discussed.

Step 1: Zero kinetic energy TOF correction   The first step is to adjust the TOF of all particles such that the particles with no kinetic energy (ending up in the centre)
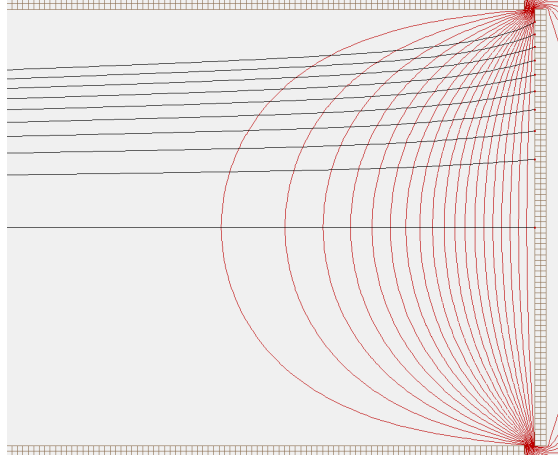
13

Figure 10: Ray traces showing the non-uniform field around the detector due to the voltage mismatch. In this case, the drift tube is kept at -4 kV and the detector front at -2 kV

have the right time of flight. The correction is done by shifting the measured TOF with an aomunt that is determined from analytical prediction and SIMION simulations. The driving force for the deviation is the difference between the detector and the drift tube voltage. Therefore, we investigate the TOF difference as a function of this difference. We define the following unit-less variables:

$$V_{nd} = \frac{V_{det} - V_{drift}}{V_{created} - V_{drift}} \tag{1}$$

$$TOF_{nd} = \frac{TOF - TOF_{noKE}}{TOF_{noKE}} \tag{2}$$

in which V_{det} is the detector front voltage, V_{drift} is the drift tube voltage and $V_{created}$ is the electrostatic potential in which the particle is created. TOF is the measured time of flight and TOF_{noKE} the time of flight of a zero-energy particle, without the lens abberation. This is the variable we are after. Note that if $V_{det} = V_{created}$ ($V_{nd} = 1$), the charge particle has no net gain in energy from source to detector. Therefore, around this voltage configuration the TOF abberation is expected to change considerably. Indeed, a steep asymptote-like behaviour is observed around this point in Figure 11.

The variables V_{nd} and TOF_{nd} have a fixed relation, independent of:

- absolute drift tube voltage

- mass of particle under study

14

- field strength in the source region

This relation is approximated by a fitted curve. Since the curve shown in Figure 11 shows behaviour similar to the natural logarithm function, a following polynomial fitting curve is used:

$$TOF_{nd} = p_2 \cdot (ln(1 - V_{nd}))^2 + p_1 \cdot (ln(1 - V_{nd})) \tag{3}$$

with p_i the fitting parameters. For the shown example, $p_2 = -1.230e-2$, $p_1 = -1.193e-3$.
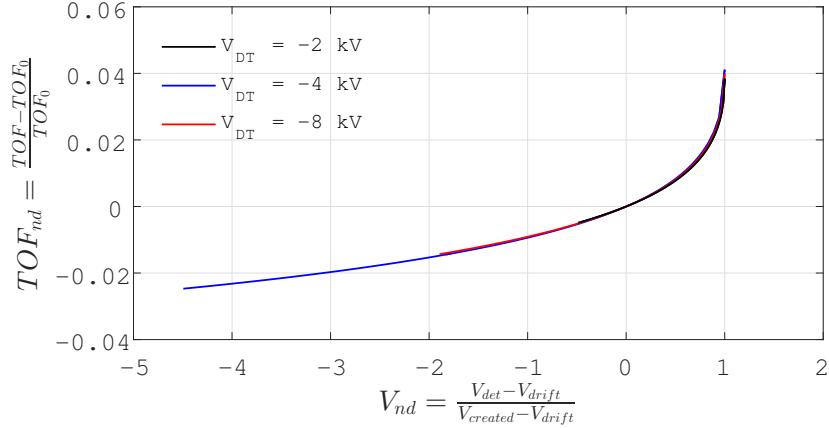
Figure 11: The Voltage and TOF dimensionless variables plotted for different absolute drift tube voltages. Similar behaviour for varying particle masses, and extraction field strengths.

Step 2: TOF correction as a function of radius   The TOF correction described in step 1 is applied to all hits. However, hits with different splat radii see a different voltage profile along its trajectory, and will therefore be affected differently. We define the following nondimensional variables:

$$R_{nd} = \frac{R}{R_{det}} \tag{4}$$

$$TOF_{nd} = \frac{TOF_{noKE} - TOF_{corr}}{TOF_{corr}} \tag{5}$$

in which R is the splat radius, R_{det} the radius of the detector, TOF_{noKE} the corrected time of flight for zero-kinetic energy particles and TOF_{corr} the corrected time of flight including the radial correction. This is the variable we are after. The

difference in time of flight appears to increase in a quadratic fashion, see Figure 12. This dependency is, again, independent of absolute voltages, particle masses and interaction field strength.
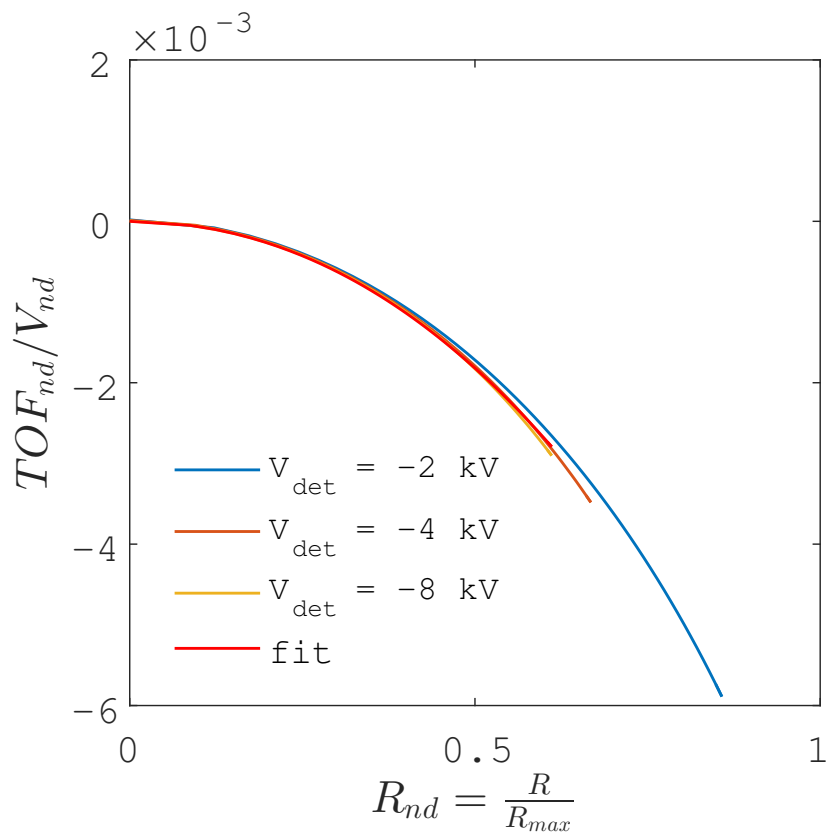
Figure 12: The difference in TOF due to a difference in field distortion at different radii. In this example the curve is fitted with $p_2 = -6.32e-3$ and $p_3 = -1.88e-3$.

The curve is fitted with the following polynomial:

$$\frac{TOF_{nd}}{V_{nd}} = p_2 \cdot (R_{nd}))^2 + p_3 \cdot (R_{nd}))^3 \tag{6}$$

The shown correction is validated by comparing it with the theoretical mass to charger conversion factor and the one with and without the correction. The theoretical value is around 965. Without correction, a factor of about 971.7 is found. After correcting, this factor is around 965, so very close to the theoretical value.

Step 3: Correction of splat radius linearity   Now that the time of flight is corrected for all hits, the splat radius will be corrected. The splat radius is expected to increase linearly with the transverse momentum vector magnitude, and this step corrects the splat radius such that it does. The example trajectories in Figure 10 show an outward curve, implicating that the corrected splat radius needs to be smaller than the registered one. This behaviour will be the other way around when the voltage difference is flipped. Therefore, the behaviour is expected to be related to the voltage difference. We assume a inverse proportional relation:

$$\frac{R - R_{corr}}{R_{det}} = p_1 \cdot V_{nd} \cdot R_{nd} \tag{7}$$

in which R is the measured radius and R_{corr} the corrected value. In the example shown in figure 15, $p_1 = 0.278$.
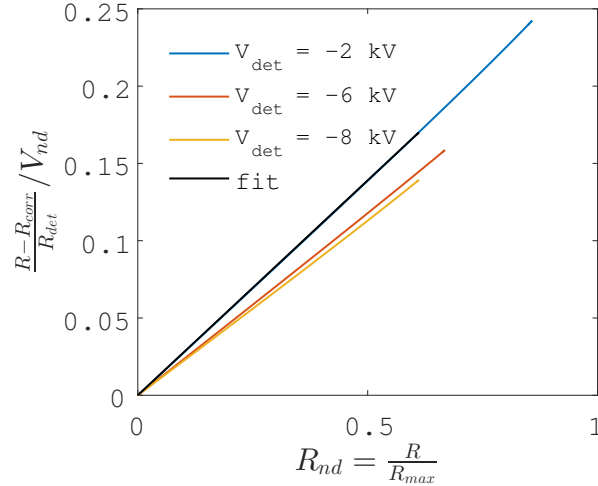
Figure 13: The splat radius correction needed, as a funtion of detector front voltage.

Metadata parameters used by the detector abberation correction   .

```
exp_md.corr.det1.ifdo.detectorabb = true; % Does this data need
    detector-induced abberation correction?
exp_md.spec.volt      = 0;    %[V] used voltages on electrodes;
exp_md.det.det1.Front_Voltage  = -2000; % [V] Detector front potential.
exp_md.det.det1.max_radius    = 40 ; %[mm] Radius of the detector
```
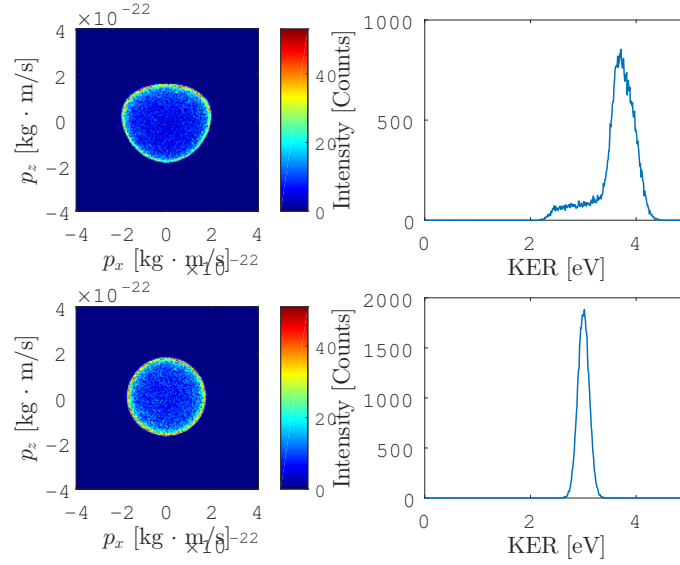
Figure 14: The momenta histograms $(\vec{p_x}, \vec{p_z})$ and KER histogram without detector abberation correction (top) and with correction (bottom). SIMION simulations, m = 18 a.m.u, singly charged, KE = 3 eV, σ(KE) = 0.1eV, 50000 particles. Drift tube @ -4kV, detector front @ -2kV.

```
exp_md.corr.det1.detectorabb.TOF_noKE.p_i = [-111 -1242 0]*1e-5; % The
    polynomial fit parameters for the TOF correction, making all
    zero-kinetic energy TOF's equal to the one without abberation.
exp_md.corr.det1.detectorabb.TOF_R.p_i   = [-1.88 -6.32 0 0]*1e-3;% The
    polynomial fit parameters for the radial TOF correction
exp_md.corr.det1.detectorabb.dR.p_i    = [0.28 0]% The polynomial fit
    parameters for the radial correction
```

TOF lens abberation correction   The TOF of a charged particle is not independent of its transversal momentum. In other words; the TOF of a particle with zero kinetic energy is different from the TOF of a particle that has an initial momentum perpendicular to the spectrometer axis. This is a direct consequence of the use of non-uniform fields (lensing). Correcting for this abberation is necessary for the KER determination. We choose a similar approach as presented before, by choosing non-dimensional parameters. Note that this correction is very similar to the one described in 'step 1' and ''step 2' of the detector abberation correction.

Step 1: Zero kinetic energy TOF correction We define the following dimensionless variables:

$$V_{nd} = \frac{V_{lens} - V_{drift}}{V_{created} - V_{drift}} \qquad (8)$$

$$TOF_{nd} = \frac{TOF - TOF_{noKE}}{TOF_{noKE}} \qquad (9)$$

$$(10)$$

in which V_{lens} is the lens voltage, V_{drift} is the drift tube voltage and $V_{created}$ is the electrostatic potential in which the particle is created. TOF is the measured time of flight and TOF_{noKE} the time of flight of a zero-energy particle, without the lens abberation. This is the variable we are after.
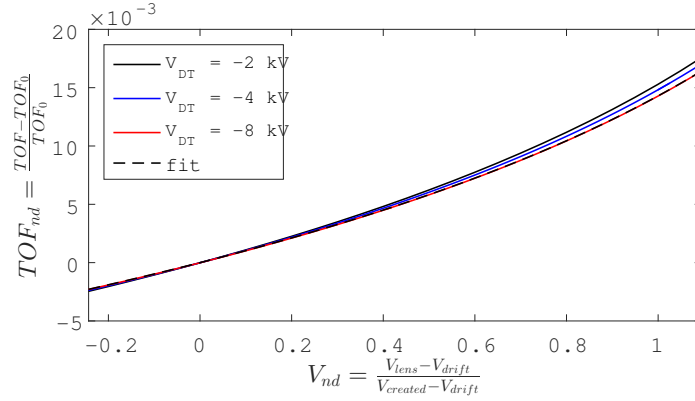
Figure 15: The Voltage and TOF dimensionless variables plotted for different absolute drift tube voltages. The fit shown uses the parameters $p_3 = 1.13e - 3$, $p_2 = 3.48e - 3$ $p_4 = 10.4e - 3$

The variables V_{nd} and TOF_{nd} have a fixed relation, independent of:

- absolute drift tube voltage

- mass of particle under study

- field strength in the source region

The relation is fitted by a mean-square polynomial fit:

$$TOF_{nd} = p_3 \cdot (V_{nd}))^3 + p_2 \cdot (V_{nd}))^2 + p_1 \cdot (V_{nd}))^1 \qquad (11)$$

Step 2: TOF correction as a function of radius Not implemented yet

19

Step 3: Correction of splat radius linearity The linearity of the radial splat with increasing velocity gets a bit lost with the use of the lens. The following dimensionless variables:

$$V_{nd} = \frac{V_{lens} - V_{drift}}{V_{created} - V_{drift}} \tag{12}$$

$$R_{ratio} = \frac{R_{corr}}{R} \tag{13}$$

The fit is shown in Figure 16.

Figure 16: The linearity correction due to the lens abberation.

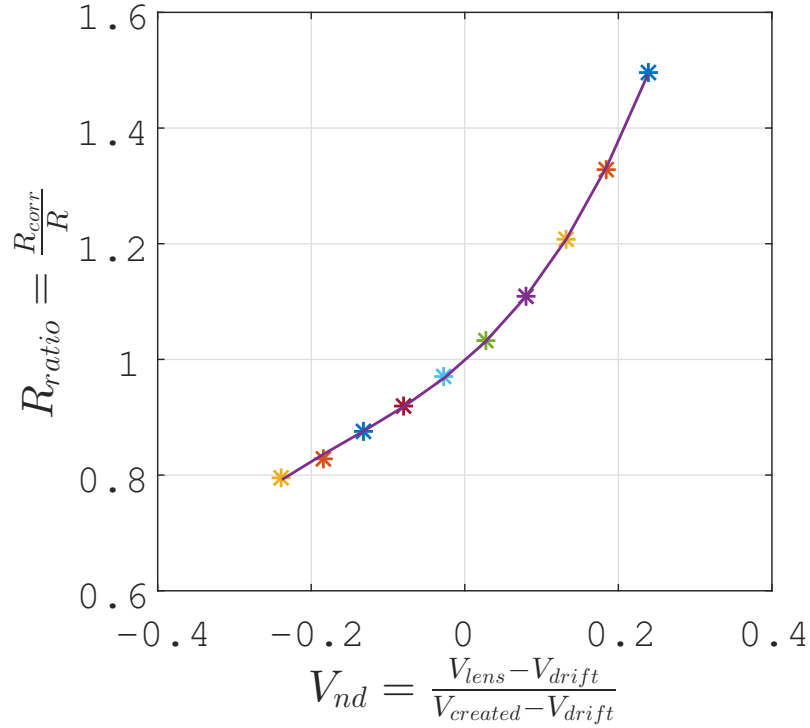It is tested and verified that the corrections can be applied to the same data, so the detector correction first, followed by the lens abberation correction.

6.2.5 Non-circular correction

It can be the case that a spectrometer shows non-circular patterns in the data, where a purely circular pattern is expected from physical principles. There is a
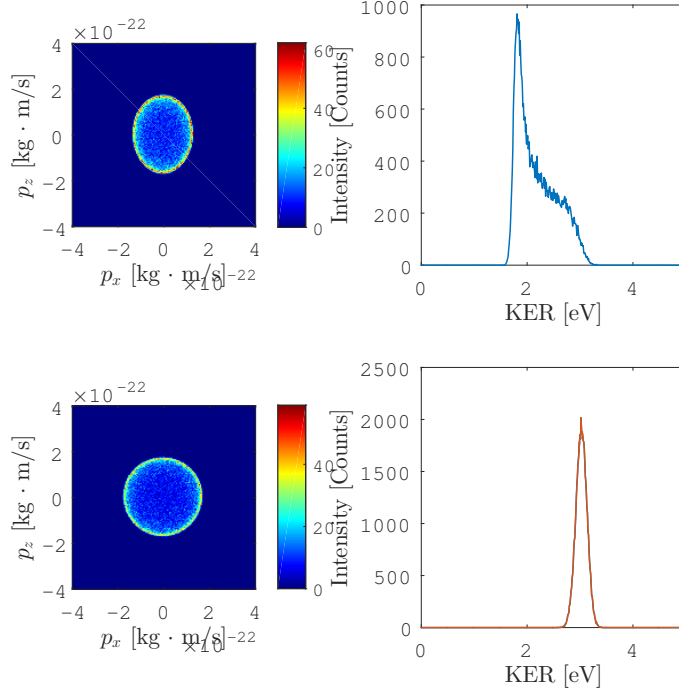
Figure 17: The momenta histograms $(\vec{p_x}, \vec{p_z})$ and KER histogram without lens abberation correction (top) and with correction (bottom). SIMION simulations, m = 18 a.m.u, singly charged, KE = 3 eV, σ(KE) = 0.1eV, 50000 particles. Drift tube @ -4kV, lens @ -3270V.

routine developed that detects a maximum along the angle in a 2D histogram with the angle (X) and another physical property (Y).

An example of such a conversion can be found in Figure 18, the result of a measurement at the EPICEA spectrometer, at the toroidal energy analyzers. The routine calculates a maximum intensity within specified radial regions, called the 'Region Of Interest' (ROI), see the black encircled boxes in Figure 18 left. The maximum-finding procedure uses median filters to the histogram in both angle (phi) and physical quantity (in this case, R) direction, to obtain a maximum line that is smoother and does not get distorted by outliers.

At zero radius, the data points will never be shifted. It is thus important to first calibrate the detection centre correctly. To prevent interpolation out of range,

the relative shift at the radial peak value is applied at a radius twice the radial peak value.

The radial values are shifted in a linear interpolation between the known points from the calibration. The interpolation is thus only applied in radial direction.
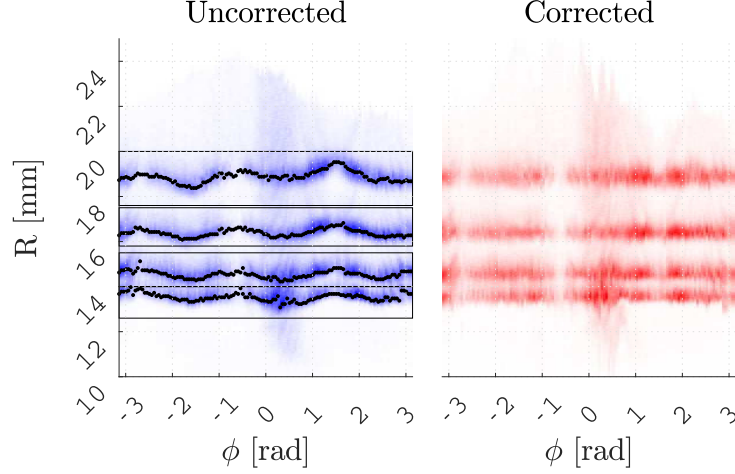
Figure 18: Left: Histogram showing the angle-dependent electron radial position from a toroidal energy analyzer. In this analyzer, the radial position is converted to electron kinetic energy. The non-circular abberation causes the oscillatory pattern, making the radius to electron energy conversion ambiguous. Right: the corrected electron radial position, showing a horizontal stripe pattern, so that a radial position corresponds to a electron energy. The location of the suspected cause of the instability (physical holders in the spectrometer) are indicated in the right plot. Data: photolines upon C1s ionization of trifluoroacetate ('ESCA molecule') at 411 eV photon energy.

6.3 convert

6.3.1 TOF → m/q

6.3.2 Signal labeling - single coincidence

By repeating a certain filtering operation onto the same signal, but with changing filtering conditions, an array with a set of labels can be written. In this example we only discuss the labeling of TOF hits to 'Mass-over-charge' values.

For example: A TOF between 950 and 1000 ns is understood as hits that originate from a mass2charge value of 1, and a TOF between 4080 and 4120 ns is understood to come form a mass2charge value of 18. A label array contains expected mass2charge values at the corresponding hits that are identified to belong to that mass2charge. There will be broadening of any TOF peak, but we can label all peaks around a certain expectation value. This broadening is called the 'search radius' in this code. Signal labeling can be done for an array of expected mass2charge values, resulting in an array with a label for every hit. No identification of a hit will result in NaN.

The definition of the conversion factor is as follows:

$$TOF = t_0 + factor * \sqrt{(M2Q)} \tag{14}$$

Note that the t_0 is a correction value and is already substracted from the measured TOF in the 'correct' section.

6.3.3 Signal labeling - double coincidence

In the case of a double coincidence event (two hits registered on one detector in one event), the filtering conditions of the measurement can be fine tuned. In this software, two methods are available; the 'circle', in which a certain maximum distance between the nominal m2q-values are approved. The other method, 'line', approves all hits along a diagonal line representing a constant TOF.

circle   In a two-dimensional histogram of two mass-over-charge values, only hits within a certain circle are approved. This can be of use in the case of cluster measurements, where the momenta of two cations are not necessarily related. See Figure 20 for an example.

Figure 19: example of a 'circle' filter around the nominal mass-over-charge values. Circle radius of 0.25 mass-over-charge units. Example of water-ammonia clusters.

line In a two-dimensional histogram of two mass-over-charge values, only hits with a TOF-sum that is close to the nominal one, is approved. This is useful when the detection is complete, and can therefore be regarded to have opposite momenta in z-direction.
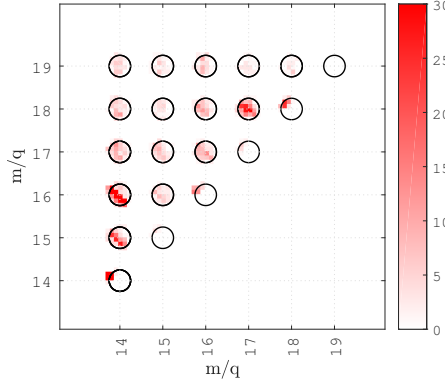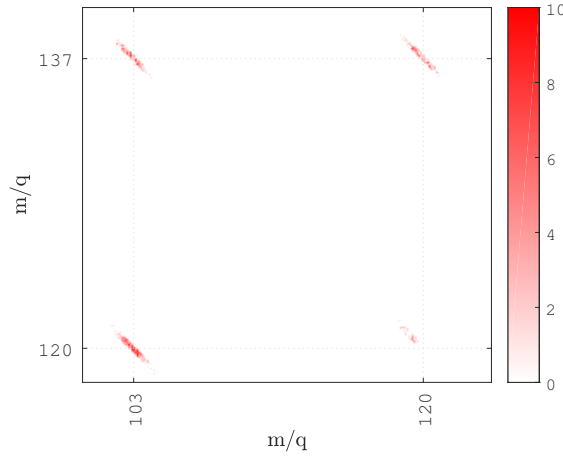

Figure 20: Example of a 'line' filter around the nominal mass-over-charge values. Circle radius of 0.2 mass-over-charge units. Example of ammonia clusters.

6.3.4 Branching ratio's

The braching ratio is defined here as the relative number of hits registered for a label, or a set of labels. In the presented example, branching ratio's of mass-over-

24

charge labels are shown.

Single-label Branching ratio's   The Branching ratio of single labels simply counts the number of hits are categorized into a certain label. As an example, figure 22 shows the number of occurences of particles recognized as having certain mass-2-charge values. Note that these labels can be defined independently from the events.
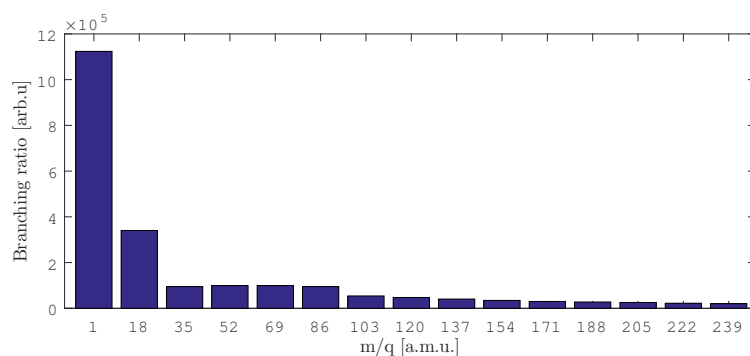
Figure 21: Example of a branching ratio plot. This example: NH_3 clusters, filename called '20091115_NH3_n17'

Double-label Branching ratio's   If multiple hits are registered in one event, the possible combinations of labels can be studied. If there are n labels registered, then n^2 combinations of labels can be registered. The double-label branching ratio is the histogram of these combinations, with a differentiation between label A registered before label B and label A registered after label B. In Figure 21, an example of such two-label branching ratio plot is shown. It shows which mass-to-charge particle is registered in combination with another. All the first labels registered have smaller mass-to-charge values, because the labels are based on the TOF values of the hits.
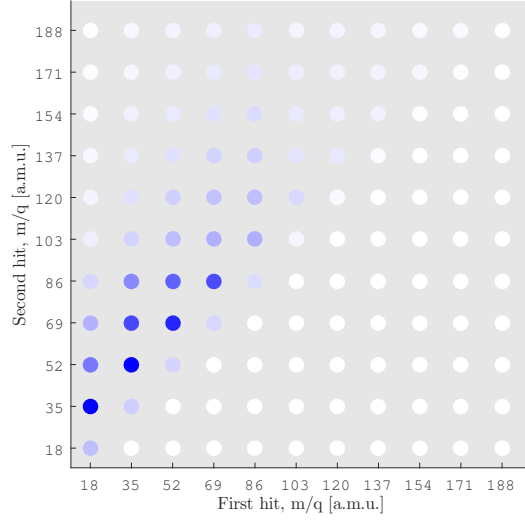
Figure 22: Example of a branching ratio plot for double labels (double coincidence). The color of the dots represents the branching ratio of that set of labels. This example: NH_3 clusters, filename called '20091115_NH3_n17'

6.3.5 Momentum

TODO: describe the Maxwell-Boltzmann distribution assumption!

$$V_{avg} = \sqrt{\frac{\gamma}{\gamma - 1}} \sqrt{\frac{2RT}{W}} \tag{15}$$
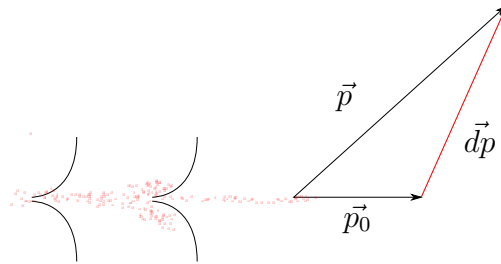
Figure 23: Schematic of the Momentum along the molecular beam, $\vec{p_0}$

In the case of momentum imaging, the splat radius on the detector gives info about the momentum in the transverse direction (here called the x and y-direction.

In the first approximation, the splat radius increases linearly with the momentum in each direction.

The momentum in the detector axis can be inferred from the difference between the actual and the zero kinetic energy time of flight of the particle.

The sample, leaving the molecular beam, has a low mean velocity in directions perpendicular to the beam. However, the velocity distribution in the molecular beam direction has a higher mean value. This will show up in the data as a spatial shift of the detection centre at higher Times Of Flight.
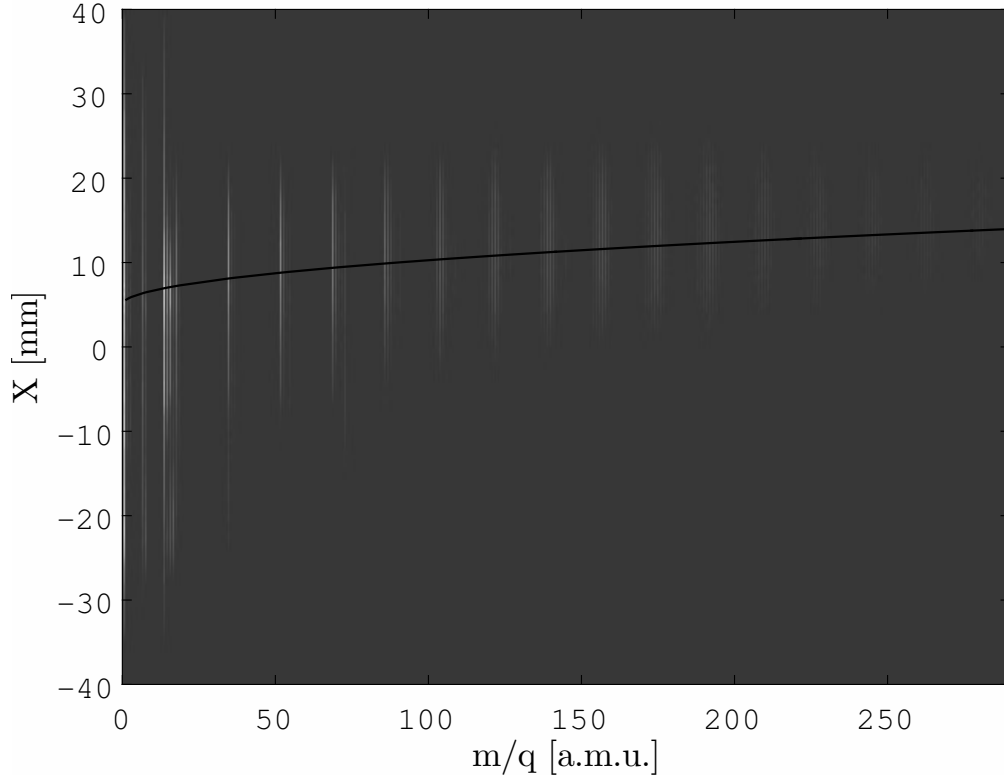
Figure 24: Radius vs TOF, showing the shift of hits outward. The line is plotted with a prediction from a Maxwell-Boltzmann expectation value of the velocity (T_{sample} = 38 [deg]). Example: Ammonia/Water clusters at 450 eV. (20130903_008.dlt)

The momentum in x and y-direction (parallel to detector plane) can be calculated from the simple equation $s = v \cdot t$:

$$\vec{p}_{x,y} = \frac{m \cdot X}{TOF} \tag{16}$$

The momentum in z-direction (perpendicular to detector plane) is calculated

as:

$$\vec{p}_z = q \cdot E_{ER} \cdot \Delta TOF \tag{17}$$

where ΔTOF is defined as the difference between the zero-momentum Time Of Flight and the actual Time Of Flight. E_{IR} is the electric field strength in the interaction region. We define two different momenta: the momentum of the sample before the photon interaction takes place ($\vec{p_0}$), and the momentum after the interaction has taken place ($\vec{p}$). Note that the kinetic energy released following photon interaction should calculate with the difference of these two vectors:

$$KE = KE(\vec{p} - \vec{p_0}) = \frac{1}{2m} \cdot |\vec{p} - \vec{p_0}|^2 \tag{18}$$

$\vec{p_0}$ **determination**   The zero-time momentum $\vec{p_0}$ is determined from the difference in the mean value of the X, Y and TOF and their zero-momentum expectation values. These zero-momentum expectation values are 0 for X and Y (assuming a proper correction of the detection centre, see signal correction), and given by the expected mass over charge labels for the TOF signal. The averaging is done over all hits belonging to one label. For instance, a label could be an OH^+ fragment.

$\vec{p}$ **determination**   The final momentum $\vec{p}$ is determined by the difference in measured detector position and the zero-momentum expectation values. Note that this value can be different for every hit, whereas $\vec{p_0}$ is determined for every label (containing multiple hits).
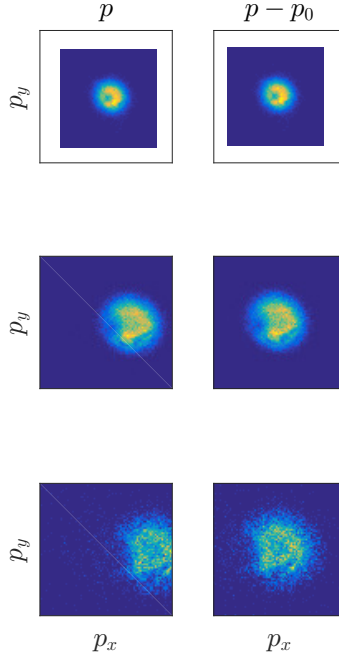
Figure 25: The total momentum (p) and the total minus the zero-time momentum ($p - p_0$) compared. In this example, the molecular beam is directed along the x-axis, coming from the left. It is seen that the mean value of total momentum shifts along the molecular beam axis, which is Example of pure NH3 clusters for cluster sizes of n = 1, 5, 10 in $(NH_3)_n^+H$

6.3.6   Kinetic Energy Release

If all three momentum components of a particle can be determined, the kinetic energy of that particle can be calculated:

$$KE = \frac{1}{2m} \cdot \left(p_x^2 + p_y^2 + p_z^2\right) \tag{19}$$

29
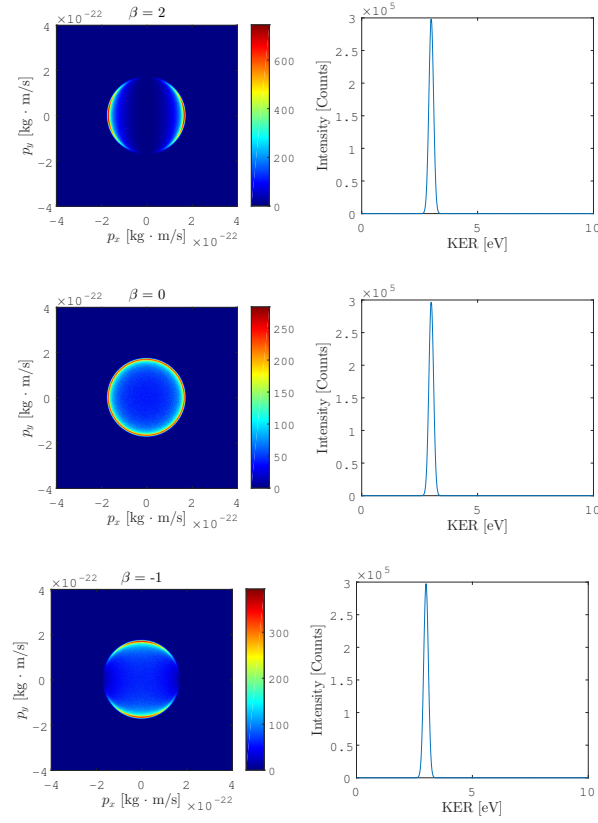
Figure 26: The kinetic energy release does not change for different angular distributions (β-values). Fake data created with the parameters: m = 18 a.m.u, singly charged, KE = 3 eV, $\sigma(\text{KE}) = 0.1\text{eV}$. Polarization is horizontal in the screen.

6.3.7 Mutual momentum angle

the angle between two three-dimensional momentum vectors is measured by the shortest great circle path between them, which means that it must lie between 0 and pi radians.
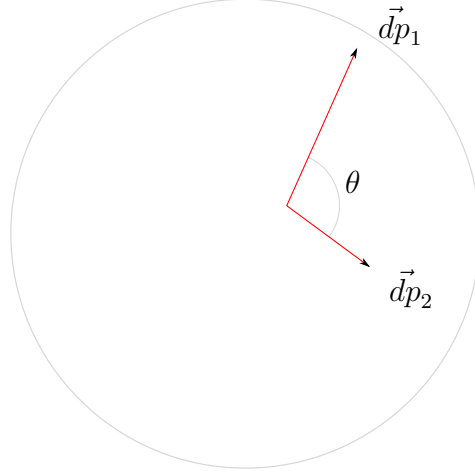
Figure 27: The shortest great circle path between two three-dimensional momentum vectors is called θ.

6.3.8 Asymmetry parameter β

TODO

6.3.9 Interaction point

In the event of complete ion detection of an initially cold sample, the initial source point position can be calculated. If we assume the sum of all momenta to be equal to zero:

$$p_{res,x} = \sum_i \frac{m_i \cdot (X_i - X_0)}{TOF_i} = 0 \tag{20}$$

where i is the amount of hits registered in one event. Form this equation, we can extract information on the creation point of the mother particle. This equation can in general be solved for any i:

$$X_0 = \frac{\sum_i \frac{m_i}{TOF_i} X_i}{\sum_i \frac{m_i}{TOF_i}} \tag{21}$$

In the case of a single-hit event, equation 21 simplifies to $X_0 = X_1$, so in these approximations, the registration of mother ions images the source volume. Equation 21 is valid for X and Y.

For the Z-direction (TOF), we identify three contributions to the measured TOF:

$$\underbrace{TOF_{exp}}_{\text{measured TOF}} = \underbrace{TOF(Z=0, p=0)}_{\text{nominal TOF, or } TOF_0} + \underbrace{\Delta TOF(Z=0, p=p_z)}_{\text{p-induced TOF shift}} + \underbrace{\Delta TOF(Z=Z_0, p=0)}_{\text{Z-induced TOF shift}}$$

$$(22)$$

The TOF difference induced by the momentum can be written as (see Eq. 17):

$$\Delta TOF(Z=0, p=p_z) = \frac{p_z}{q \cdot E_{ER}} \qquad (23)$$

The TOF difference induced by the Z-displacement is the relation under study here. Note that this number is minimized by the Wiley-McLaren condition, so this calculation can be error-prone when exact Wiley-McLaren conditions are met. We assume the linear condition:

$$\Delta TOF(Z=Z_0, p=0) = \frac{C \cdot Z_0}{\sqrt{m/q}} \qquad (24)$$

In case of Wiley McLaren conditions met, C = 0 (see Figure 28), and the presented treatment cannot be used.
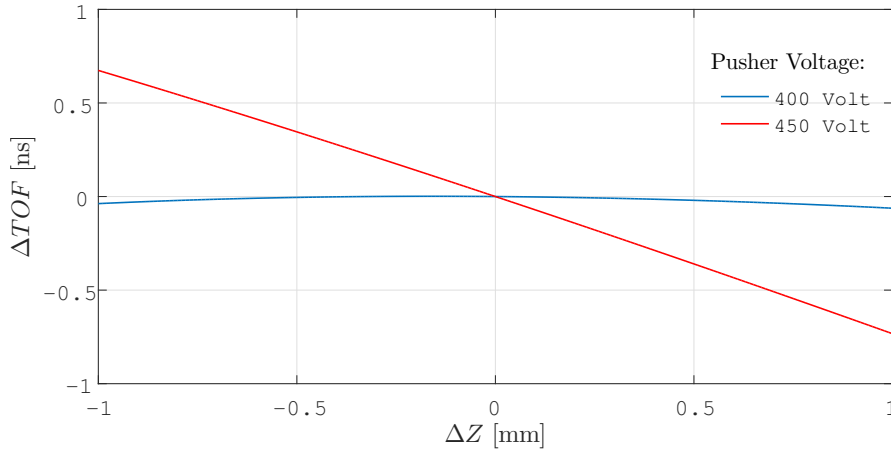

Figure 28: Example of how the change in dZ affects the effective (TOF Laksman). Two examples of ion 'pusher' voltages are shown. The 400 Volt example meets the Wiley-McLaren condition, but the 450 Volt example does not. In this treatment, the correlation is assumed linear. $m/q = 1$, Drift tube voltage = -4kV.

With this description of the TOF, we can use momentum conservation:

$$p_{res,z} = \sum_i p_{z,i} = E_{ER} \cdot \sum_i \left(TOF_{exp,i} - TOF_0 - \frac{b \cdot Z_0}{\sqrt{m/q_i}} \right) \cdot q_i = 0 \qquad (25)$$

$$\sum_i \left(TOF_{exp,i} - TOF_0 \right) \cdot q_i = \sum_i \frac{b \cdot Z_0}{\sqrt{m/q_i}} \cdot q_i \qquad (26)$$

Note that Z_0 is an event property, and is thus the same for all hits in the same event. To write Z_0 explicitly:

$$Z_0 = \frac{\sum_i \left(TOF_{exp,i} - TOF_0 \right) \cdot q_i}{\sum_i b \cdot \sqrt{\frac{q_i^3}{m}}} \qquad (27)$$

Or, more general, not assuming a linear relation between the ΔTOF and Z_0:

$$a + b \cdot Z_0 + c \cdot Z_0^2 + ... = \frac{\sum_i \left(TOF_{exp,i} - TOF_0 \right) \cdot q_i}{\sum_i \sqrt{\frac{q_i^3}{m}}} \qquad (28)$$

This conversion is only implemented for X and Y. See 'convert.source_position' for the code so far.

6.3.10 Oversight of signals

Here, a signal is defined as something that has a value for all hits. New signals are created by correcting a raw signal or converting a corrected signal. The signals can be found in: *'Experiment name'.h.'detector name'.'signal name'*. The momentum is expressed in atomic units. [1]

Description	MATLAB name	Type	Created from	Unit
detector dependent	raw	input	-	-
corrected X	X_corr	corrected	X (raw)	mm
corrected Y	Y_corr	corrected	X (raw)	mm
corrected TOF	TOF_corr	corrected	TOF(raw)	ns
mass-to-charge	m2q	converted	TOF_corr	$\frac{a.m.u}{a.c.u.}$
m2q labels	m2q_l	label	m2q	$\frac{a.m.u}{a.c.u.}$
mass labels	m_l	label	m2q	a.m.u
zero-time momentum, $\vec{p_0}$	p_0	converted	X, Y, TOF, m2q_l	a.u.
final momentum, $\vec{p}$	p	converted	X, Y, TOF, m2q_l	a.u.
momentum difference, $\vec{p} - \vec{p_0}$	dp	converted	X, Y, TOF, m2q_l	a.u.
Kinetic energy release	KER	converted	p, p_0	Joule

---

[1] ($1 a.u. = 1.992e - 24 \cdot \frac{kg \cdot m}{s}$)

Table 1: Input and output array for a filter only approving input values between and including 4 to 9. ($4 \leq x \leq 9$)

input array	output array
3	false
4	true
7	true
2	false
12	false
8	true
1	false

6.4 filter

In this context, filtering is defined as checking whether single hits or events holds a certain quantitative condition or criterium. No data is thrown out. Thus, the output array of a filter contains boolean (values are 'true' or 'false'). An example of an input signal filter, its input and output matrix, is given in the following table.
combining multiple filters can be done in a single line.

- *filter_array_3 = filter_array_1 && filter_array_2* if both filter conditions should hold

- *filter_array_3 = filter_array_1 ||filter_array_2* if either condition should hold.

Filtering can be done on the events and on the hits. An example of a filter on events is the filtering of all double coincidence events. An example for hits is a TOF that filters out a certain chemical mass. When a set of hits is filtered, the corresponding hits can be identified and thus a event-based filter array can be built. This can be done with function XXXX. This can be needed when a certain condition is imposed on one detector, and the user wants to visualize the remaining events on another detector.

The filtering arrays are stored under the name *filter_h_*name* for events, and *filter_h_*det_name* _ *name* for hits. For the hit filter arrays, the detector name needs to be specified, since hits among different detectors need not be correlated.

Events

Event nr	Det 1	Event filter
1	1	true
2	3	false
3	NaN	true
4	6	true
5	NaN	false
6	8	true
7	9	true
8	11	false
.		

Hits

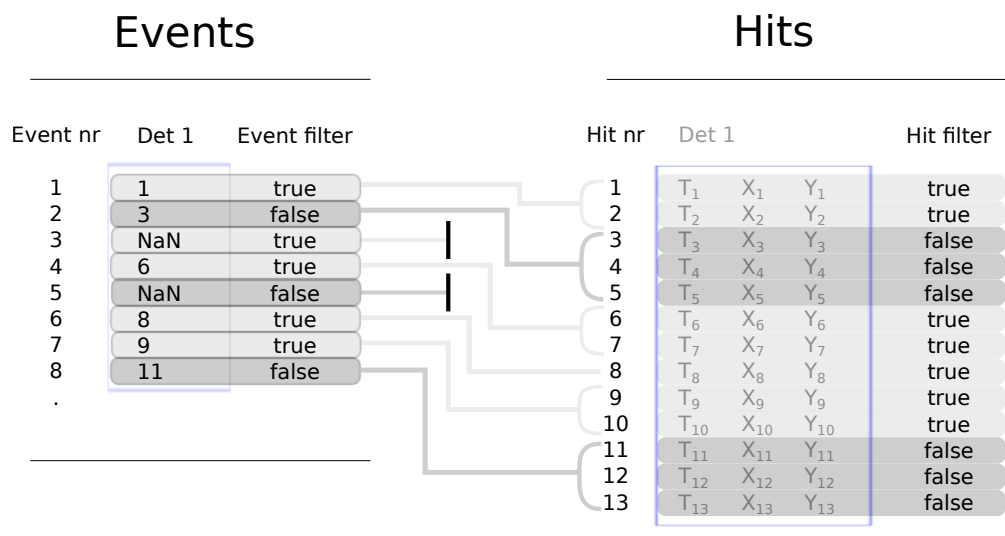| Hit nr	Det 1			Hit filter
1	T_1	X_1	Y_1	true
2	T_2	X_2	Y_2	true
3	T_3	X_3	Y_3	false
4	T_4	X_4	Y_4	false
5	T_5	X_5	Y_5	false
6	T_6	X_6	Y_6	true
7	T_7	X_7	Y_7	true
8	T_8	X_8	Y_8	true
9	T_9	X_9	Y_9	true
10	T_{10}	X_{10}	Y_{10}	true
11	T_{11}	X_{11}	Y_{11}	false
12	T_{12}	X_{12}	Y_{12}	false
13	T_{13}	X_{13}	Y_{13}	false

Figure 29: The translation of an example event filter to a hit filter.

6.5   plot

TODO: The available plot functions and a few examples of signals plotted.

6.5.1   Mass to charge

6.5.2   momentum

Hits

Hit nr				Hit filter
1	T_1	X_1	Y_1	true
2	T_2	X_2	Y_2	false
3	T_3	X_3	Y_3	true
4	T_4	X_4	Y_4	false
5	T_5	X_5	Y_5	false
6	T_6	X_6	Y_6	true
7	T_7	X_7	Y_7	true
8	T_8	X_8	Y_8	false
9	T_9	X_9	Y_9	true
10	T_{10}	X_{10}	Y_{10}	false
11	T_{11}	X_{11}	Y_{11}	false
12	T_{12}	X_{12}	Y_{12}	true
13	T_{13}	X_{13}	Y_{13}	false

Events

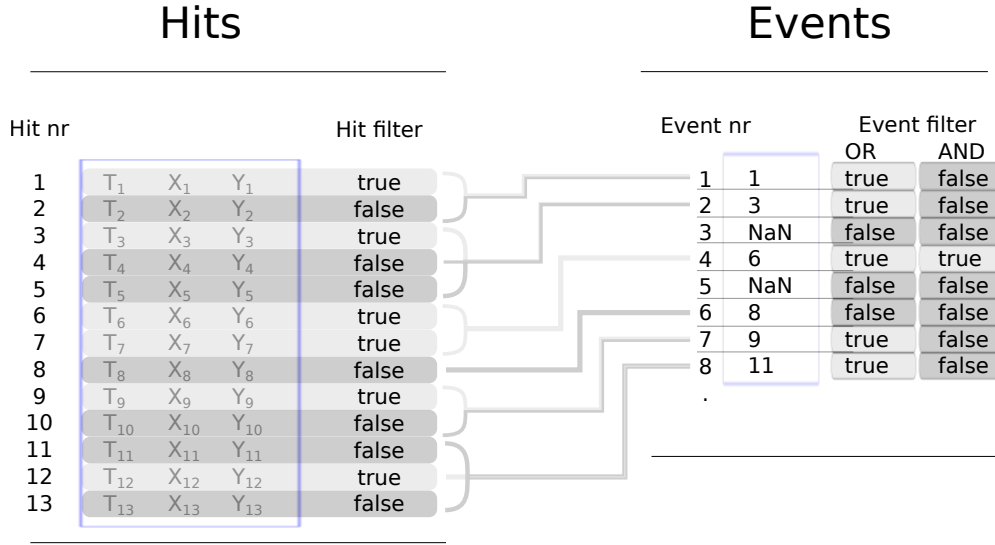| Event nr		Event filter OR	AND
1	1	true	false
2	3	true	false
3	NaN	false	false
4	6	true	true
5	NaN	false	false
6	8	false	false
7	9	true	false
8	11	true	false
.			

Figure 30: The translation of an example hit filter to an event filter.

6.6 calibrate

The calibrate functions go through a procedure that return correct correction and conversion factors. These should afterwards be stored in the metadata, category 'conversion factors'.

TOF $\rightarrow m/q$ **factor**

detection center position

detector rotation

residual momentum minimualization The residual momentum of a two-body process, resulting in two cations, can be used to calibrate the spectrometer. This module minimizes the relative norm of the residual momentum:

$$f_{min} = \frac{norm(\vec{p}_{res})}{norm(\vec{p}_1 - \vec{p}_2)} = \frac{norm(\vec{p}_1 + \vec{p}_2)}{norm(\vec{p}_1 - \vec{p}_2)} \tag{29}$$

The variables that are tweaked during this optimization:

- corr.det1.dX (X-shift of the raw image)

- corr.det1.dY (Y-shift of the raw image)

- spec.volt.Ve2s (voltage of grid between electron MCP and source ('pusher'))

- conv.det1.TOF_2_M2Q.factor (Conversion factor from mass 2 charge to TOF)

- conv.det1.TOF_2_M2Q.t0 (time correction)

This function requires the input values to be roughly calibrated, so should be used as a final calibration: the solver only searches in a small solution domain and only a local minimum is found.
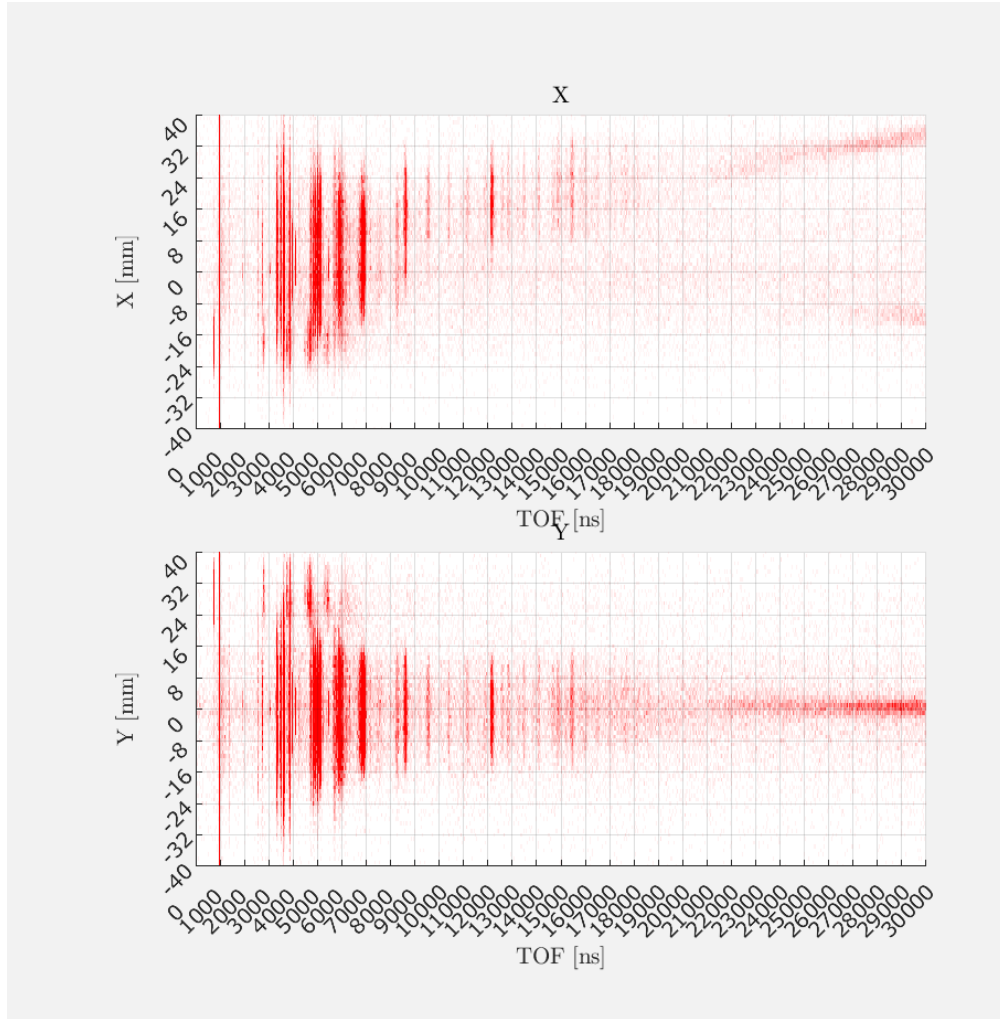

Figure 32: Example of a molecular beam calibration

6.7 general

Functions of general use are stored in this category, that are outside of the standard package of MATLAB. These can be written for the package, and of use for all package categories. They can also originate from MATLAB File Exchange (http://nl.mathworks.com/matlabcentral/fileexchange/) The package has no parameters or constants stored, they are in general all provided by the metadata (see section 'metadata structure'). An exception is the general constants of nature, that are more or less written in stone. For example the unit conversion factors (converting eV to Joule). These constants are also stored in this category.

```matlab
% Example of MATLAB codes/functions
[input1, input2] = function test(input1, input2)
```

7 Installation

The package is installed by adding the directory of the package folder to the MATLAB path.
In order to use Erik Mansson's import software, the path to the folder containing the struct definition needs to be added as well.
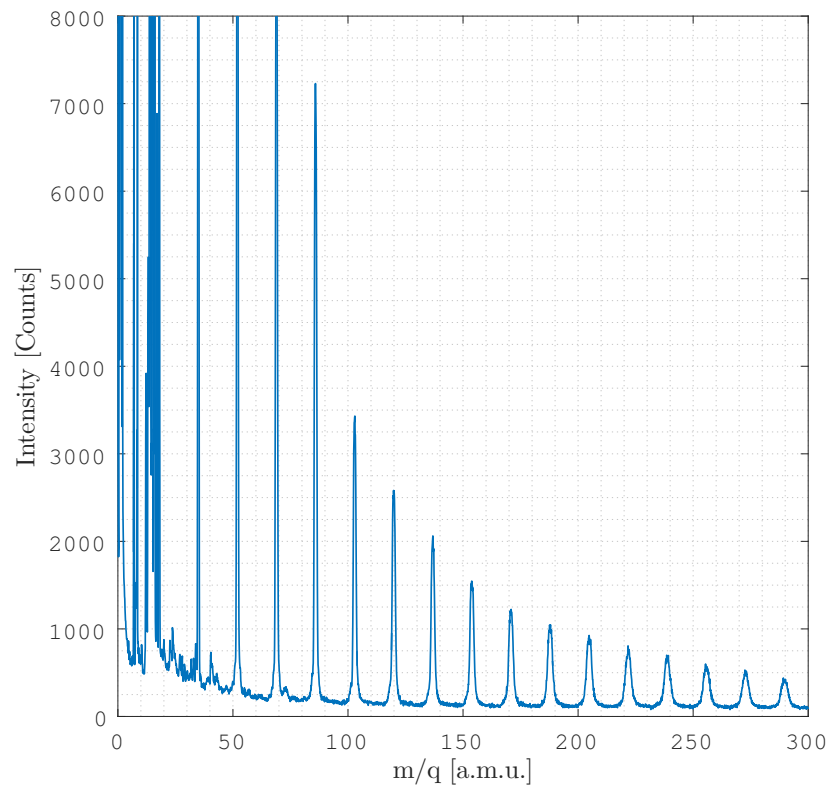
Figure 31: Example of a mass-to-charge histogram of pure NH$_3$-clusters from a molecular beam.