AlphaTims graphical user interface tutorial

Developed by: Eugenia Voytik, Sander Willems and Matthias Mann.

This step-by-step guide helps you to get started with our software AlphaTims.

Table of contents

Program description

AlphaTims is an open-source Python package for fast accessing of unprocessed Bruker trapped ion mobility spectrometry - time of flight (TIMS-TOF) data. It provides a very efficient indexed data structure that allows to access the five-dimensional TIMS-TOF data in the standard numerical Python (NumPy) manner. AlphaTims is a key enabling tool to deal with the large and high-dimensional TIMS-TOF data.

Installation

- There are several types of installation for the AlphaTims package:

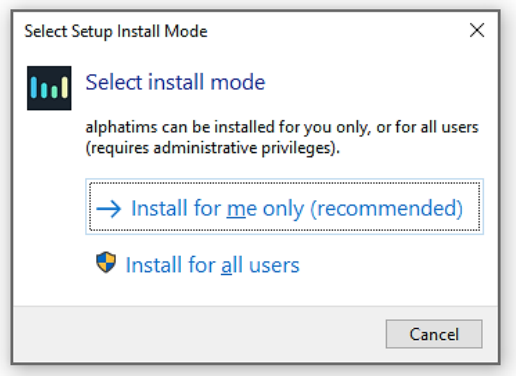
- a graphical user interface(GUI) installation for users that don't have enough experience in Python programming. This installation will be completely covered in this tutorial.

- a Python installation as a separate package using pip or a full installation of the editable version of the whole package. About these types of installation and how to use the package in Python you can read in more details in the GitHub repo.

Windows

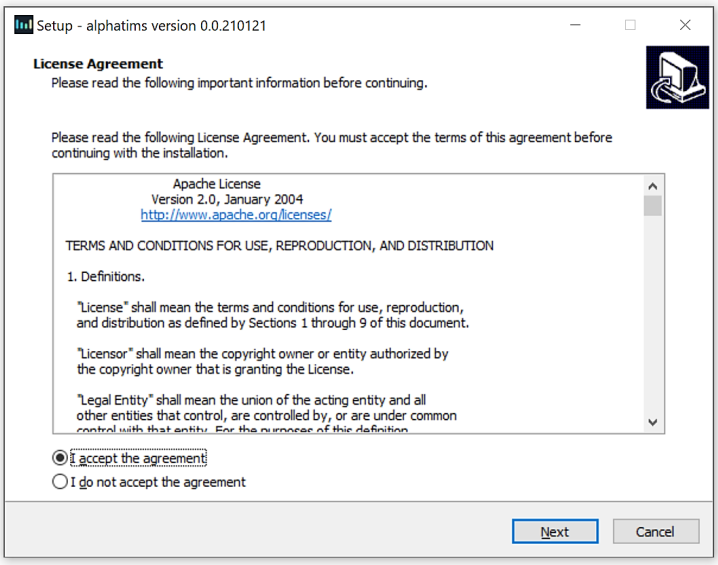
1. Download the latest release for Windows (alphatims_installer_windows.exe) from the GitHub repository and open the .exe file.

2. In the appeared "Select Setup Install Mode" dialog we suggest to select "Install for me only (recommended)" option.

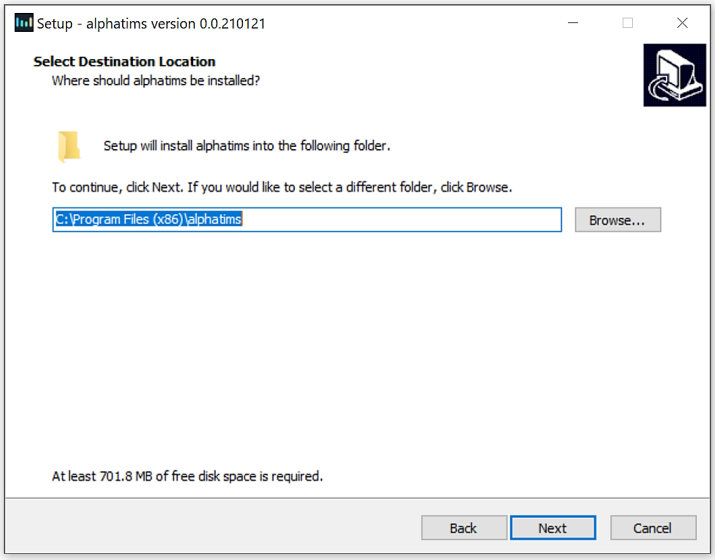


IMPORTANT WARNING! If you occasionally install AlphaTims for all users, you might need admin privileges to run it each time on your computer (right-click on the AlphaTims logo on your desktop and select "Run as administrator").

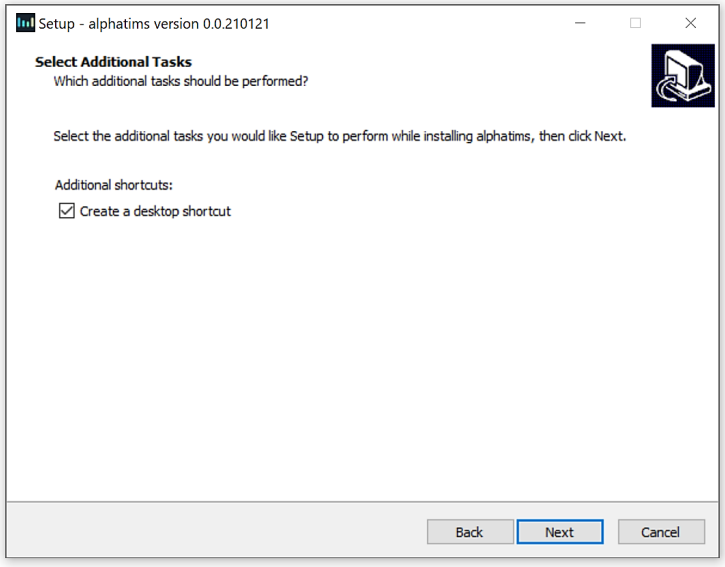
3. In the appearing "Setup – AlphaTims version X.X.X" dialog window, accept the License Agreement and press the "Next" button.



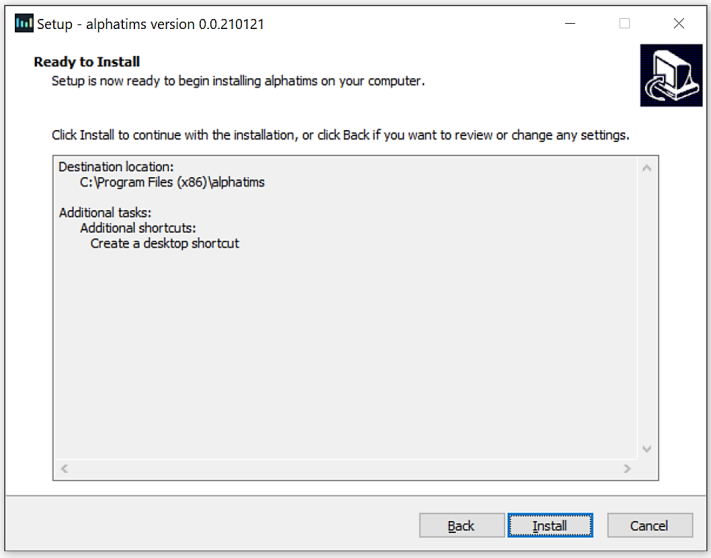
4. Select the destination location for the installation of AlphaTims software and press the “Next” button.



5. (optional) In the next dialog window, mark the “Create a desktop shortcut” check box and press the “Next” button.



6. Check the setting and if everything is correct, press “Install” button. You may go back to change some settings using the “Back” button or “Cancel” the installation.



7. The installation process is indicated by the progress bar. Wait until the installation process is finished. It is advised to directly test if the installation was successful by marking the “Launch AlphaTims” check box before pressing the “Finish” button.

8. In the appearing “Windows Security Alert” dialog window press the “Allow access” button that will prevent the Windows Defender Firewall from blocking the AlphaTims tool on your PC.

9. A new tab should have been opened in your default browser (Google Chrome or Mozilla

Firefox are suggested for the fast running of the AlphaTims). You can now use AlphaTims.

MacOS

1. Download the latest release for macOS (alphatims.app.zip) from the GitHub repository, unzip the file and move it to your applications folder. By doing so, you accept the terms of the AlphaTims license agreement and all third-party licenses.

2. Launch the file and take into account that the first opening of the tool on macOS takes a long time to load. The loading time will be significantly reduced upon the second launch.

3. A new tab should have been opened in your default browser (Google Chrome or Mozilla Firefox are suggested for the fast running of the AlphaTims). You can now use AlphaTims.

   * If you get a pop-up message “Do you want the application “python3.8” to accept incoming network connections?”, just click “Yes” button.

   * If nothing happens when you launch AlphaTims, you might need to grant it permissions by going to the macOS menu "System Preferences | Security & Privacy | General". If the problem still persists, it is possible that macOS already quarantined the AlphaTims app. It can be removed from quarantine by running <*xattr -dr com.apple.quarantine AlphaTims.app*> (copy everything between <>) in your terminal in the application folder where AlphaTims.app is located.

Linux

1. Download the latest release for Linux (alphatims) from the GitHub repository. By doing it, you accept the terms of the AlphaTims license agreement and all third-party licenses.

2. To launch the file drag-and-drop it in the terminal and press enter to run AlphaTims.

3. A new tab should have been opened in your default browser (Google Chrome or Mozilla Firefox are suggested for the fast running of the AlphaTims). You can now use AlphaTims.

* If nothing happens when you launch AlphaTims, you might need to grant it permissions by running <*chmod +x alphatims*> (copy everything between <>) in your terminal in the application folder where AlphaTims is located.

How to use AlphaTims

1. Upload your dataset:

On this step you might be interested in checking the information about your dataset manually. This can be done either opening “analysis.tdf” file in the original .d folder, e.g. using DB Browser for SQLite, or an HDF compass for already saved .hdf file. There you can easily find e.g. global metadata about the sample, all available frames and precursors, etc.

a) Provide the filepath to the original raw Bruker .d folder, or a saved .hdf file, in the “Specify an experimental file:” field, e.g. “D:\Bruker\20201207_tims03_Evo03_PS_SA_HeLa_200ng_EvoSep_prot_DDA_21 min_8cm_S1-C10_1_22476.d”.

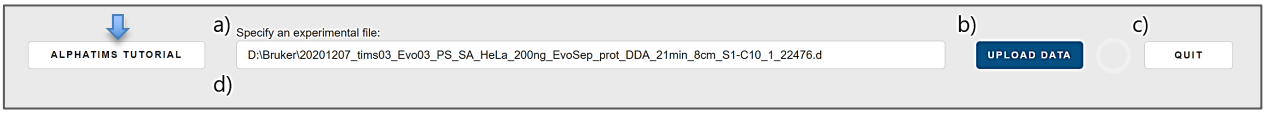
b) Press the "Upload Data" button. The loading process is indicated by a spinner symbol. To evaluate the required time to preload the file into memory take into account the typical performance statistics for the reading of raw/HDF data. Some additional time is required to load all visual components of the dashboard .

c) Please, press the “Quit” button when your working session in AlphaTims is finished.

IMPORTANT WARNING! If you just close the browser tab and do not press the "Quit" button, AlphaTims will keep running in the background (possibly using a huge amount of RAM memory). This is especially important for **macOS** because of the non-showing terminal. Therefore, you can always reopen the closed tab using a keyboard shortcut “Ctrl + Shift + T” or reopen it from the history of your browser.

d) If the specified file won’t have a .d/.hdf extension or it will be impossible to upload it for exploration because of any other reasons, you will get an error message below the “Specify an experimental file:” field with the reasons what is wrong.

Just pressing “AlphaTims tutorial” button you can always download the current tutorial from the dashboard.



* See the example files for testing of the GUI in the GitHub repo.

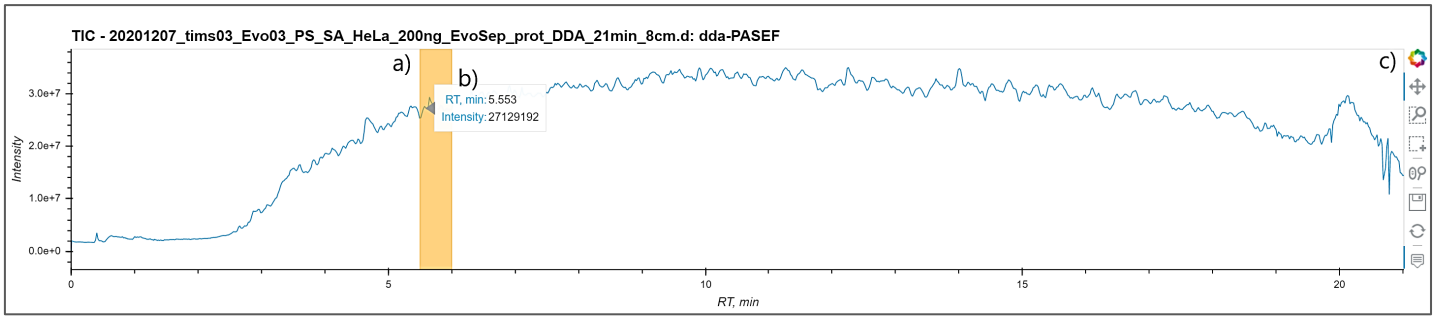
2. On the right you may see three different plots for the data exploration:

   - **TIC** - a line plot that visualize the total ion chromatogram for the whole run.

     a) The sliced and currently visualized on other plots RT range is highlighted in orange.

     b) If you hover over the chromatogram, all annotation information for the position of the cursor will be displayed, such as a precise RT in minutes and an intensity value.

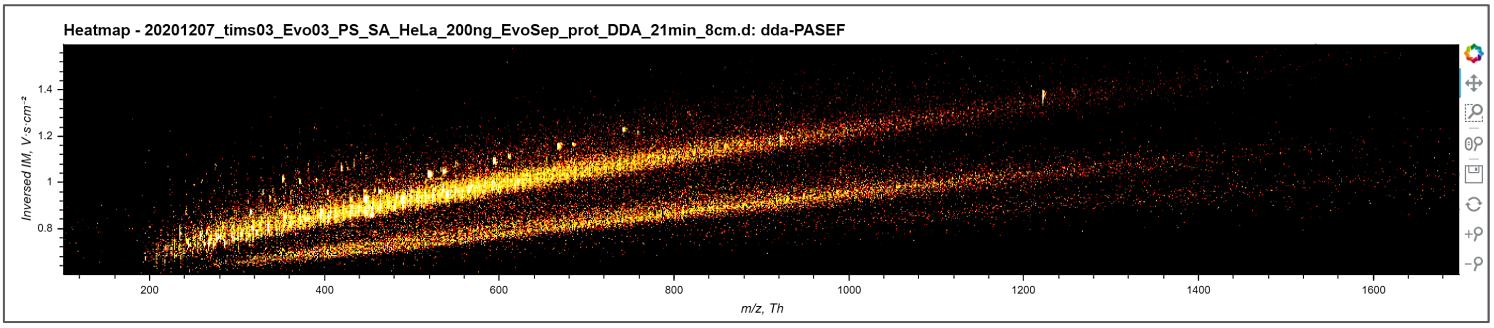
     c) Below you can find a detailed instruction about all available options of the plot's toolbar that allows to easily manipulate the plot, e.g. to zoom in and out.



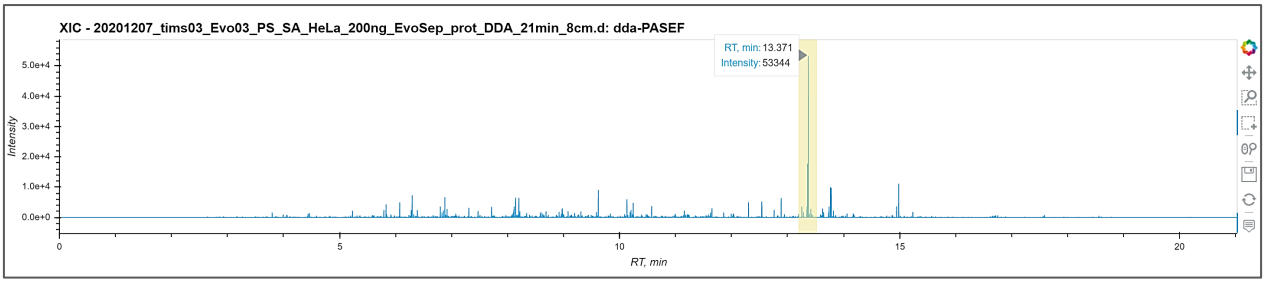
   - **Heatmap** - a scatter plot for the sliced data on the requested axis (could be "m/z, Th", "RT, min" and "inversed IM, $V \cdot s \cdot cm^{-2}$") colored by the summed intensities. This plot is generated with the use of Datashader Python library that allows to "rasterize" or "aggregate" the huge amount of data points into regular grids based on some aggregator statistics functions, like sum or mean. Thus billion points datasets can be viewed as images, making them easier to observe the internal patterns. Using interactive tools, such as zooming in/out, each plot will be rebuilt and colored based on a new amount of data. In our case a "pure white" pixel indicates the highest summed intensity of a single ion/or a group of ions.

     - Below you can find a detailed instruction about all available options of the plot's toolbar that allows to easily manipulate the plot.

     - For this plot you can change the x- and y-axis in the "Select axis for plots" card in the "Parameters" section.



- **Spectrum / XIC / Mobilogram** - a line plot for the sliced data that visualize "m/z, Th", "RT, min" or "inversed IM, V·s·cm^{-2}" against “intensity” values.
  - The sliced and currently visualized on other plots RT range is highlighted in light yellow.
  - If you hover over the plot, all annotation information for the position of the cursor will be displayed, e.g. a precise RT in minutes and an intensity value.
  - Below you can find a detailed instruction about all available options of the plot’s toolbar that allows to easily manipulate the plot.
  - For this plot you can change the x- and y-axis in the “Select axis for plots” card in the “Parameters” section.



Each plot as you could already see comes with a number of interactive tools:

-  - The pan tool allows to pan the plot by left-dragging a mouse.
-  - The box zoom tool helps to define a rectangular region to zoom the plot bounds to by left-dragging a mouse.
-  - You can use a box selection tool to define a rectangular selection region by left-dragging a mouse and as a result to highlight a region.
-  - The wheel zoom tool allows to zoom the plot in and out, centering on the current mouse location.

-  - The save tool allows you to save a PNG image of the plot.
-  - The reset tool restores the plot ranges to their original values.
-  - The hover tool allows to hover over a given data point and display its associated information.
-  - The zoom-in tool increases the zoom of the plot.
-  - The zoom-out tool decreases the zoom level of the plot.

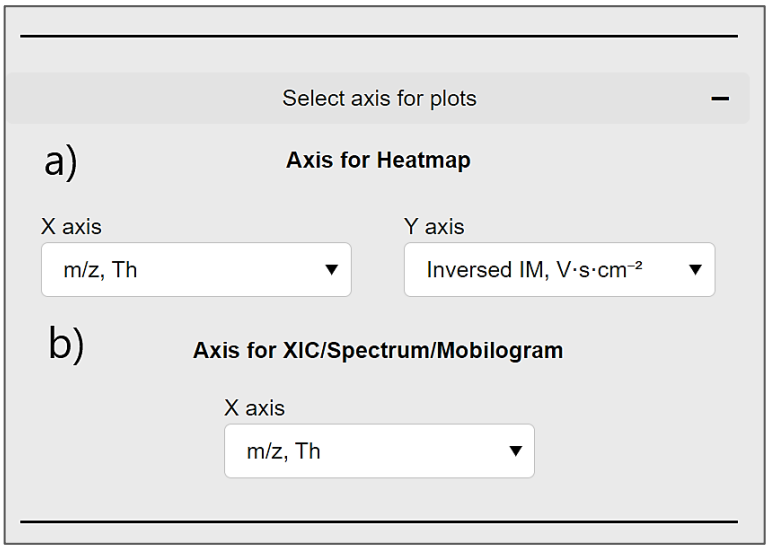
3. On the left in the “Parameters” section there are several opened/collapsed cards with options that can be specified for slicing of the original data and further visualization/exporting:

   - **“Select axis for plots” card**

     This card allows changing the axis for the two images presented in the dashboard plots. It can be done for:

     a) The scatterplot/heatmap where you can choose between three options for x-/y-axes, such as "m/z, Th", "RT, min" and "inversed IM, V·s·cm^{-2}".

     b) The line plot with the options to plot spectrum ("m/z, Th" vs. “intensity”), extracted ion chromatograms, or XIC ("RT, min" vs. “intensity”) or mobilogram ("inversed IM, V·s·cm^{-2}" vs. “intensity”).



- **LC dimension: “Select rt_values / frame_indices” card**

  a) For a quick overview of the uploaded dataset you may use an integrated player option that allows to visualize ten MS1 frames which are distributed with the same

interval through the whole gradient. The player has a number of buttons to go to the last (“⏭”) or the first (“⏮”) selected frame, step forward(“⏭”) or backward (“⏮”), or visualize 10 selected frames automatically one by one in a forward (“▶”) or a backward (“◀”) order and pause the visualization (“⏸”). It also provides control over the loop policy which determines whether to play the visualization 'once', 'loop' through it, or 'reflect' the process. Additionally “-” and “+” buttons slow down and speed up the player speed.

* Before using the player consider that it will override/ignore any custom frame settings while taking into account the settings of all other widgets.

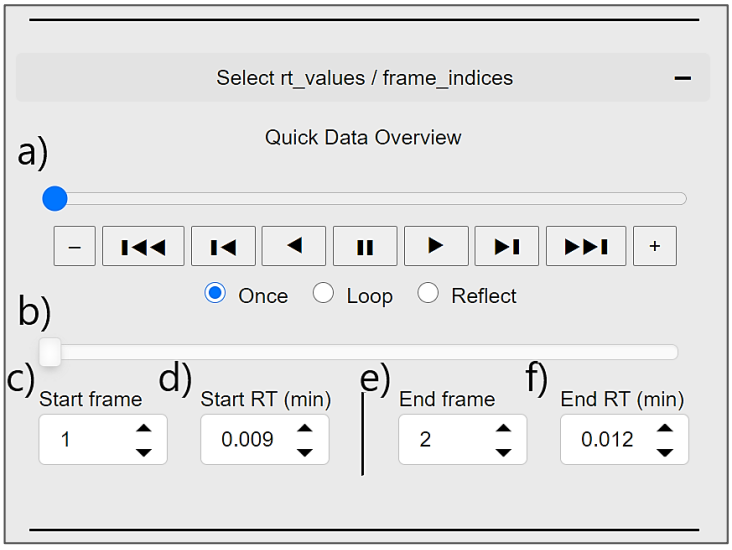
* Don’t use a high speed for the player for the complex datasets because it won’t be enough time for the plots to be updated.

b) Having a single frame or a range of frames for visualization, you may specify them using a frame range slider that shows the start and the end of desired frames. As per default in Python, the last frame won’t be included into selection.

c) – f) Simplifying the work with the slider in b) you may also set manually the start – c) – and end – e) – of the frame range or the start retention time (in minutes) – d) – and the end retention time – f) – values. For the c) – f_ fields you may also use the up- and down-arrows to set the values of these fields.

* Pay attention that manually specifying the values in the c) – f) fields they can be automatically slightly updated to match the actual values.

* If any specified start value is equal to or higher than the end value, the end value will be changed automatically with the value equal to “start_value + 1”. The same behavior will be observed when any specified end value is equal to or less than the start value.



➢ **TIMS dimension: "Select mobility_values / scan_indices" card**

This card allows slicing mobility values and scan indices. It can be done using either a scan slider and the "Start scan"/ "End scan" input fields for manual setting of the values or the "Start IM"/ "End IM" value inputs to specify the range of inversed ion mobility values ($1/K_o$). For example, setting a value for the "Start scan" field will automatically update the slider start value and will change correspondingly the value of the "Start IM" field.

* For more detailed information see the description of the **LC dimension: "Select rt_values / frame_indices" card.**

➢ **QUADRUPOLE dimension: "Select quad_values / precursor_indices" card**

This card focusses on the quadrupole and indirectly on the collision cell. It allows to decide which ions, precursor or fragments, will be visualized and to slice fragments either based on their lower and upper quadrupole m/z values or by their precursor indices.

a) To select only the precursor ions, mark the "Show MS1 ions (precursors)" check box.

b) To select only the fragment ions, mark the "Show MS2 ions (fragments)" check box.

* You may select both precursors and fragment ions just mark two checkboxes at

the same time. Take into account that it's impossible to distinguish between precursor and fragment ions in the plot.

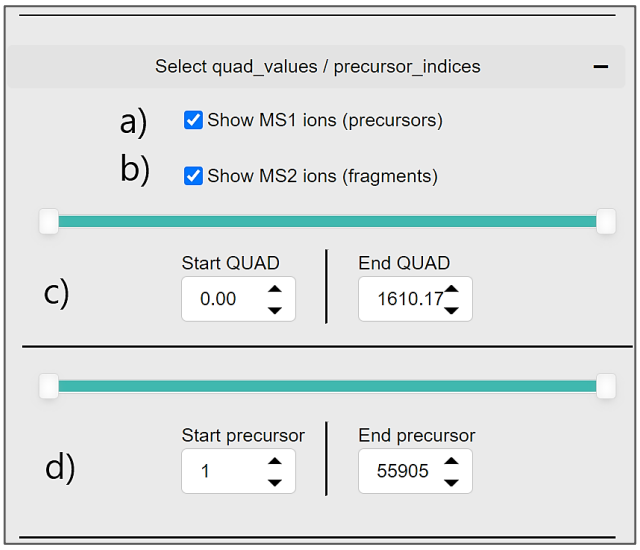
c) To slice quadrupole m/z values use either a quadrupole values slider or the "Start QUAD"/ "End QUAD" input fields for manual setting of the values.

* Take into account that as long as there is an overlap between the instrumental settings and the selected quadrupole range, the tool retains the results even if we can't determine whether they are present or not. E.g. if the instrument selects m/z values in range between 600 and 603 m/z, our filtering between 600.1 and 600.15 m/z will still retain all fragments that are acquired, even though their actual precursor is more likely to be around 601.5.

d) To slice precursor indices use either a precursor indices slider or the "Start precursor"/ "End precursor" input fields for manual setting of the values.

* Important to know what these indices exactly mean. For data dependent acquisition, by selecting a single precursor index value you can actually extract a traditional MS2 spectrum aggregated over multiple mobility scans or even frames. In turn for data independent acquisition, this refers to a frame window group. In this case a single window group in DIA actually allows for multiple different quadrupole settings, which means it can be combined with scan/mobility settings for a higher level of details. These indices for both cases can be manually found in the SQL database or in the .hdf file (see step 1).

* For more detailed information see the description of the **LC dimension: "Select rt_values / frame_indices" card.**



- **TOF dimension: “Select mz_values / tof_indices” card**

  This card allows slicing TOF m/z values and indices which are identical for precursors and fragments. It can be done using either a TOF indices slider and the “Start TOF”/ “End TOF” input fields for manual setting of the TOF indices or the “Start m/z”/ “End m/z” value inputs to specify the range of fragment m/z values. For example, setting a value for the “Start TOF” field will automatically update the TOF indices slider start value and will not linearly change to the correspondent value of the “Start m/z” field.

  * For more detailed information see the description of the **LC dimension: “Select rt_values / frame_indices” card.**

- **Intensity dimension: “Select intensity_values” card**

  This card allows filtering the data base on the intensity values of the ions. It can be done using an intensity values slider or the “Start intensity”/ “End intensity” input fields for manual setting of the intensity values.

  * For more detailed information see the description of the **LC dimension: “Select rt_values / frame_indices” card.**

- **“Export data” card**

  This card allows exporting the data into two different formats:

  - HDF format

    To reduce the reading time of your original file by 2-3 fold and, as even more important, to define an easily portable file format which can be transferred between different OSs, it can be saved into HDF5 format that are supported for reading in step 1a.

    a) Provide the path to the folder where to save a created .hdf file, e.g. “D:\Bruker\my_file.hdf”.

       * By default, the file will be saved with the same file name to the same folder where your original .d folder exists. For example, in our case it would be: “D:\Bruker\20201207_tims03_Evo03_PS_SA_HeLa_200ng_EvoSep_prot_DDA_21min_8cm_S1-C10_1_22476.hdf”

    b) If an “overwrite” check box is marked, an existing .hdf file will be modified without any options to restore its previous version. If it’s unmarked, a new file

will be created only if the specified file name doesn't exist.

If a "compress" check box is marked, a compression will be used for an .hdf file creation that will reduce the file size on the disk at the cost of taking 3-6 times longer to access the file later. If it's unmarked, no compression is used.

c) Press the "Save to HDF" button. The loading process is indicated by a spinner symbol. To evaluate the required time to preload the file into memory take into account the typical performance statistics for the export to HDF.

d) In case of the successful saving of the file or any errors a corresponding message will appear in this field.

* a) - d) options are disabled if user originally uploads .hdf file.

- CSV format

Applying all specified slicing options, e.g. frames and scan ranges, the filtered data can be saved into .csv format with all possible information about ions:

- raw_indices
- frame_indices
- scan_indices
- precursor_indices (0 value is equal to "precursor" ion)
- tof_indices
- rt_values
- mobility_values
- quad_low_mz_values and quad_high_mz_values (-1 value is equal to "precursor" ion)
- mz_values
- intensity_values

These saved data could allow you to recreate any figures or build your own visualization.

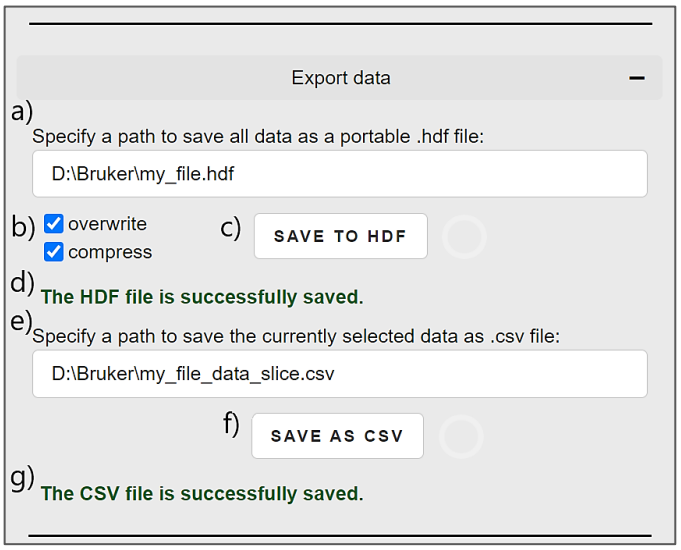
e) Provide the path to the folder where to save a created .csv file, e.g. "D:\Bruker\my_file.csv".

* By default, the file will be saved with the same file name to the same folder where your original .d folder exists. For example, in our case it would be: "D:\Bruker\20201207_tims03_Evo03_PS_SA_HeLa_200ng_EvoSep_prot_DD

A_21min_8cm_S1-C10_1_22476_data_slice.csv”

f) Press the "Save as CSV" button. The loading process is indicated by a spinner symbol.

g) In case of the successful saving of the file, you’ll get a confirmation message.



- **“Undo | Redo” buttons**

  To reverse your actions inside “Select …” cards where you are setting values for slicing/visualization of the data, you can use “Undo” and “Redo” buttons. They won’t have any effects on exporting, strike count estimation, image zooming or (re)loading other datasets.

  You can reverse more than one action just clicking several times on the buttons. You can use “Redo” command only after “Undo” command.

  * Take into account that it could take some time to update the widgets values and the plots.

- **“Strike count (limit / current estimate)” option**

This option sets an estimation of the number of data points that can be sliced at the same time. For example, by default it’s allowed to visualize till 10 million points and if you will try to set the RT ranges for the whole dataset, the data won’t be sliced and both the heatmap and the line plot will be empty. This is done for time performance considerations.