# PIMP User Guide

PIMP combines ion time-of-flight (TOF) spectra from multiple i2PEPICO measurements into a two-dimensional photoionization matrix, picks reusable mass-peak windows, and plots breakdown fractions and summed ion yields. The vertical axis can show any numeric parameter exported to the index file, including photon energy, temperature, gas flow, pressure, current, slit position, detector rate, or elapsed time.

The TOF files and index are generated by the Export2D Imports of i2PEPICO Analysis 3.3 or the i2PEPICO measurement program.

## Files to distribute with i2PEPICO

- `pim.py`: the single-file PIMP viewer, including Peaks, My traces, 3D and workspaces
- `pim.pdf`: the complete export and analysis user guide
- `requirements.txt`: the Python runtime packages
- `ScriptImport.exe` (analysis package): installs the Export2D, Export2DSLS1 and Export2DSLS2 Imports into an existing `script.dat`; the bundled analysis `script.dat` already contains them. The matching rebuilt ScriptImport also embeds a fourth analysis Import, Export2DSlices; builds dated before 16 October 2026 install only the first three
- `Export2DExample.scr`: general Process example, also present in the bundled analysis `script.dat`
- `Export2DSlicesExample.scr`: Process example that writes one closed raw file as a time-sliced folder (`Imports[Export2DSlices]`); see Time-sliced folders
- `process_h2o.scr`, `process_h2o_folders.scr`: one-folder and three-folder H2O Process examples
- `rebuild_export2d.scr`: recovery Process example
- `measure_pseudo_live.scr`, `measure_pseudo_live_simple.scr` (measurement package): full and short H2O Measure examples

Runnable templates are in the package's `Examples` folder. The analysis package contains the offline subset of those scripts, together with the same viewer and guide.

See the H2O sample scripts in `Examples` for configuration and the rebuild/watch/resume workflow. These callers belong under Measure/Process; only the library scripts belong under Imports.

The libraries belong in i2PEPICO's top-level **Imports** script folder as **Export2D**, **Export2DSLS1**, **Export2DSLS2** and, for measurement, **Export2DMeas**. There are no underscores in those import names. For analysis, close i2PEPICO, copy `ScriptImport.exe` beside your `script.dat` and double-click it, or drop your `script.dat` onto it. It adds or updates the analysis Imports (three in current builds; four, including **Export2DSlices**, in the matching rebuilt ScriptImport), keeps all other scripts and leaves a backup beside the database. Analysis and Process scripts use only `Imports[Export2D]`, which loads both schemas without any Measure command; time-sliced exports use `Imports[Export2DSlices]`, which loads Export2D itself. Measure scripts use only `Imports[Export2DMeas]`, which loads Export2D and its schemas automatically, then provides Export2DMeasureOpen and Meas. Keeping the measurement wrapper separate prevents offline imports from arming the native measurement-notification setup warning.

Export2D indices contain their own parameter names and titles. `channels.dat` is no longer used by PIMP.

## Installation

Python 3.10 or newer with Tkinter and Matplotlib 3.10 or newer are required. This version has been checked with Python 3.13 and Matplotlib 3.11. Install the additional packages with:

```
python -m pip install -r requirements.txt
```

Start the viewer with:

```
python pim.py
```

`requirements.txt` installs the required runtime packages. Peaks, My traces, 3D, workspace JSON, icons and multi-folder selection introduce no additional runtime packages. The upward-swimming fish icon is embedded as PNG and a multi-resolution Windows ICO. PIMP sets its own Windows app identity before opening a window. Tkinter comes from the Python/operating-system installation rather than pip; on a Python installation without Tk, install its platform's Tk support. Check the installation with `python -c "import tkinter, numpy, matplotlib"`. The python.org installers for Windows and macOS include Tkinter. On Windows, the `py` launcher can replace `python` in these commands; on macOS and Linux the command is usually `python3`. Homebrew Python on macOS needs `brew install python-tk`.

# Preparing and recovering the fixed export folder

Each raw sample/scan folder has one fixed `PIMP` child, containing the canonical `2D_TPES_index.txt` and three TOFs per indexed measurement. For example, `C:\Data\H2O\PIMP` receives the results for raw files in `C:\Data\H2O`. The fixed child name is defined only in `Export2DFolders` in `export2d.scr`.

To recreate a baseline from already closed raw files, paste `rebuild_export2d.scr` as a **Process** script. Select the raw folder in the input controls, check the schema, mask and ring, then run the script:

```
Imports[Export2D]
Ring := ROIRing[(0;0),0.03,0.06,1]
SaveMyself[CurrInFolder]
Gosub[Export2DRebuild, CurrInFolder, *.i2d, ROIRingAsCircle[Ring], Ring]
```

SLS2 is the default; for historical SLS1 files, add `UseSLS1 := 1` after Imports in this Process script. The ROI values are examples and must match the experiment. The default `*.i2d` mask selects all matching files in the raw folder without recursion, including H2O coarse, fine and monitor files. A narrower mask is available. No matches raises the ordinary CycleFiles error before opening a replacement index. Rebuild replaces the canonical index and overwrites the selected standard TOFs; rerunning it produces the same baseline row count. Raw measurements and unrelated results are retained. PIMP follows indexed files; leftover spectra cannot create extra measurements. Ambiguous spectra for an indexed record require cleanup.

Run rebuild while acquisition/export is idle and **stop PIMP monitoring first**. It overwrites spectra before publishing the index; these files are not one transaction. After rebuilding, Stop then Play performs a full reload. An interrupted rebuild can be rerun while idle.

The low-level `Export2DScan[Mask, RawFolder, OutFolder, Center, Ring]` still supports an explicitly chosen destination. `Export2DOpen` replaces an index; `Export2DResume` publishes pending output before checking whether the canonical index exists, then appends and writes the header only when it is absent. Both initialize schema, TOF settings and coincidence criteria. Close before choosing another session. Existing headers are not compared with current settings: consistent Center/Ring, schema, columns and TOF settings on resume are the caller's responsibility. Rebuild is the remedy for changing that baseline.

# Point-by-point acquisition

New acquisition uses SLS2.

Paste measure_pseudo_live.scr as a **Measure** script and adapt its ROIs, durations, beam-current check and energy ranges. Type or select the intended day-relative sample in **Meas**, for example `H2O` or `H2O\Run1`. Its two scans use `H2O.1!!!.i2d` and `H2O.2!!!.i2d` and share one PIMP session. Optional fixed-energy references use `H2O.1.monitor!!!.i2d` and `H2O.2.monitor!!!.i2d`. Set `MonitorEnabled := 0` to omit references while retaining live updates and beamline monitor columns. `FileCounter` comes from each `For` loop: it starts at zero in each scan, and each reference shares its scan point's number. The shorter measure_pseudo_live_simple.scr keeps one 13-point scan from 12.40 to 13.00 eV in 0.05 eV steps, with 60-second measurements named `H2O!!!.i2d`, no reference measurements and no beam-current wait. Choose either template according to the acquisition you need.

Every runnable example uses `SaveMyself` to preserve the executed script: measurement and rebuild examples save in the raw folder, and offline export examples save in each output folder after exporting it.

Saving requires that destination to be writable. See the H2O sample scripts in `Examples` for the complete workflow.

These are main scripts, started with the ordinary Measure/Process controls. Only the four libraries belong in Imports. The interpreter inserts an End boundary before each imported library and flushes output at script completion; callers without local helpers need no final End, FlushOutput or Close. The H2O measurement caller has an End before its local monitor helper to prevent fall-through. Use Close only when switching export sessions within one run. `CurrFullSample` gives the complete absolute folder of the sample currently typed or selected in Meas, with a trailing separator, and creates it if necessary; `CurrSample` returns the GUI sample field. Native Measure accepts either an absolute folder or a short name under the configured day folder.

**Export2DMeas acquisitions stay in the current day's sample tree.** The raw folder passed to Export2DMeasureOpen and Meas must lie under the configured day folder (`Root\YYYY-MM\YYYY-MM-DD\`), and neither it nor the data root may contain spaces. The data root must use an ordinary drive-letter path, such as `D:\Data\`, rather than a UNC or device path. Choose the root before starting and keep it unchanged until the script finishes. Use `CurrFullSample`, as the templates do. A relative raw-folder argument is resolved from the working directory, not from the day folder, and is normally rejected. Meas converts the checked raw folder to its day-relative sample with the native `DayRelativeSample` command and passes that to native Measure, so the Meas field keeps showing the relative sample (e.g. `acetaldehyde\Run1`) instead of a full path. A folder outside the day, such as a previous day's full path still shown in Meas, is rejected before the wrapper creates raw/PIMP folders; choose a valid day-relative sample before restarting. Previously saved absolute sample values are not automatically remapped to today. Offline Export2D processing is not restricted.

Install the rebuilt measurement `i2PEPICO.exe` containing `DayRelativeSample` and replace and save the contents of the existing **Imports / Export2DMeas** node with the matching `export2dmeas.scr`; update both. Copying a `.scr` file into a scripts directory does not update the installed Imports node. Keep the local `script.dat` and its other scripts.

`Meas` is a Gosub label with exactly four actuals: raw folder, filename pattern, duration and minimum beam current. It uses native Measure's default Refresh, copies `LastMeasuredFile` after the raw file closes, processes that exact resolved path, writes Center/Ring/All TOFs and one index row, then requests publication. It checks the active raw-folder association before acquisition and stops with a local event-log explanation if the session is missing or belongs to another folder. Changing folders requires an explicit Close/Open transition. Repeated MeasureOpen for the same active folder preserves that session and its ROIs.

Another script run resumes the same canonical index. Rebuilding the old baseline and subsequently measuring one new point is supported; orphan raw files are backfilled only when the user deliberately rebuilds. Native Measure has no hard-coded exporter hook. `LastMeasuredFile` returns an absolute resolved path only after successful file-backed completion; failed, memory and no-destination attempts clear it. Loading an older file does not change it.

Export2D owns the one native Output destination while active. Do not redirect Output to a log during its session. Append uses `Output[Folder, Name, 2]` and native `FileExists`, without shell commands or header reconstruction. An existing empty/malformed index is preserved. Failed temporary-file replacement remains ignored, and final Stop/error cleanup can publish partial pending text. PIMP uses only the canonical index, never `.tmp`. Aborted measurements are not automatically indexed; the user may select retained closed raw files for recovery.

## Electron regions

Set the Ring before every export and derive its matching Center:

```
Ring := ROIRing[(-0.107;-0.144),0.025,0.05,1]
Center := ROIRingAsCircle[Ring]
```

The coordinate pair is the ring center, followed by its inner radius, outer radius and anisotropy. ROIRingAsCircle returns a circle with the same center, inner radius and anisotropy; it leaves the ring unchanged. Update the ring when the electron image position or size changes.

The coincidence criteria must remain in the order Center, Ring, All. The export assigns `TOFRes[1]`, `TOFRes[2]`, and `TOFRes[3]` to `_center`, `_ring`, and `_all` respectively.

### Extended index

The script creates `2D_TPES_index.txt` with the measurement file name, acquisition time, measurement start time, and numeric i2PEPICO parameters. The main groups are:

- photon energy, temperatures, Keithley current, and MFC flow rates
- experimental, source, gas-filter, sample, roughing, and foreline pressures
- ring current, detector/channel rates, and image rates
- horizontal and vertical slit positions
- selected status values, including GF Mix and GF Ne as numeric values
- voltages, currents, ramp values, and other diagnostic channels

The schema imports define channel IDs, names, units and their fixed order. Export2D obtains parameter values with `LoadedParam[Start, channel ID]` and writes `LoadedMeasTime` and `LoadedMeasStart` separately. Missing parameters stay empty and are read as unavailable by PIMP; they never shift later columns. Filenames, parameter names and title matching are case insensitive. The first spelling of an unknown parameter is retained for display.

Do not reorder only the header or only the values. Check the schema import channel IDs against the acquisition configuration when that configuration changes.

The export folder contains the index and matching `_center.tof`, `_ring.tof`, and `_all.tof` files.

## Using the viewer

Choose **+** above the folder list to open **Add export folders**. Select the actual `PIMP` child of a scan, not the raw scan directory. A raw directory is only a place to browse. The dialog also lists each immediate scan's `PIMP` child, labelled by the scan name, so several scans under one parent can be selected together. Ordinary export folders require `2D_TPES_index.txt`; named PIMP folders can be opened while still empty. Ctrl-click on Windows/Linux or Command-click on macOS selects individual PIMP folders; Shift-click selects a range. Double-click a directory to browse inside it. **Up**, the editable path, and **Browse...** change the location. **Browse...** selects an opened export directory and enables **Add selected**, which accepts it. The current directory is selectable when it is named `PIMP` or contains the canonical index. The picker lists one directory level at a time and does not search recursively. Adding folders leaves any active monitor unchanged. Repeat **+** to add another group of folders without replacing the analysis. The picker starts in the folder used last (remembered across sessions), else next to the last loaded folder, else in the folder that contains `pim.py`.

The whole accepted group is composed and drawn once. Duplicate directories are skipped. If some folders fail, compatible folders are added and a single report names the rejected folders and their reasons; no rejected folder is loaded. Cancelling the picker, or a group with no acceptable folders, leaves the current data and applied settings intact.

The left column lists loaded folders by scan name. Two scans with the same name show the shortest preceding path that distinguishes them, such as `day1/Scan_17` and `day2/Scan_17`. Hover a name for the full PIMP path. Check or uncheck a folder to include or exclude its rows without rereading its files. Excluded names and scales are muted; scales remain editable, and a pending or invalid multiplier keeps its own border. Clicking a name also toggles its checkbox, indicated by the hand cursor and hover help. The active watch target is marked **Watching** in teal and stays visually separate from the inclusion checkbox. The **-** button immediately removes the viewer's source record; no data files are deleted. Adding the same directory again does not duplicate its measurements. Folder names wrap to the available pixel width, and the column grows with the window. Mixed additions report skipped duplicates as well as added/rejected folders. The toolbar above the folder list holds, from left to right, **+** (add folders), **Open workspace**, **Save workspace**, **Peaks**, **My traces**, **3D view** and **Export**; hover over an icon for its name. Peaks and My traces look pressed while their window is open. Without data, 3D view and the data exports are disabled and the 3D tooltip says why. Ctrl+O opens and Ctrl+S saves a workspace, and Alt+A opens the Export menu, from the main, Peaks, My traces or 3D window. Close the main window to exit. Use the wheel or scrollbar when the list is long.

Loading reports the current folder and measurement count in the status area. Loading, workspace operations, 3D creation, matrix rebuilding and numerical exports show a busy cursor in open analysis windows. Progress is repainted without dispatching queued user actions. Work still runs on the Tk thread, so this is progress feedback. Automatic watch loads run in a worker while Tk remains responsive.

Opening or browsing into an indexed export directory selects that directory and enables **Add selected**. Offline export folders are recognized by their canonical index even when the directory is not named `PIMP`. Add one or more indexed folders with **+**. The dialog only adds folders; it does not start monitoring. Folders named `PIMP` may also be added before their index or first measurement exists.

## Watching completed measurements

Add folders with **+**. Each folder named `PIMP` has **Play** and **Stop** buttons between its Scale field and remove button. Its source name is the parent scan name, with enough path context to distinguish identical names. Other indexed export folders can be opened normally and have no monitoring controls.

**Play** starts monitoring that folder and disables every Play button. Only the active folder's Stop button is enabled. **Stop** retains its loaded data and settings, stops polling, and enables Play on every PIMP folder. There is one active monitor per PIMP instance. Monitoring is independent of source inclusion and does not stop acquisition. Opening another folder does not switch the target.

An empty or header-only PIMP folder appears immediately as a waiting card, with inclusion and Scale disabled until its first complete measurement loads. Play and Stop work before that first measurement. The first accepted data populate the same folder identity, without adding a duplicate source. Waiting cards are saved in workspaces as explicitly pending source records; reopening restores all cards with monitoring stopped. Removing a card keeps its files and forgets its monitoring target if active. The source marker says **Watching** while monitoring; the status panel gives waiting, update or retry details and its full-path tooltip.

Polling defaults to five seconds, followed by two seconds of settling after an index change. No index or a header-only index shows a normal waiting state. Missing folders, temporary read errors and incomplete new spectra show inline unavailable/retry information. The last valid data stay displayed and the same candidate is retried, even if its timestamp is unchanged. Only one read job can run; changes are coalesced and obsolete results are discarded.

**Play** rereads every indexed spectrum when monitoring starts, including old files deliberately overwritten without changing index rows. Unavailable spectra already accepted by an offline load remain historical gaps; new records must have all three complete spectra. Restarting can fill a historical gap once its file is available. A second File header is rejected before rows can be associated with metadata. For a rebuild, press **Stop**, rebuild while acquisition/export is idle, then press **Play**. Removing its source, replacing a workspace or exiting invalidates pending work.

Ordinary growth loads only new indexed spectra and reorders cached rows. Equal coordinates remain separate records. **Select latest** defaults off and appears inside the active watched source entry, beneath **Watching**. It is hidden when monitoring stops or fails, and disabled when that source is excluded. Its preference is retained while hidden. Checking it neither starts monitoring nor selects an existing measurement: each subsequent accepted arrival becomes the explicit measurement selection. When unchecked, selection stays with its source UUID/index-row identity, and unfinished field/peak-editor text survives updates. For the same scan coordinate, including a change that only alters its capitalisation, and, for elapsed time, the same time origin, each x and y range of the matrix, mass spectrum, ion yield and an open My traces window is decided separately: a range still at its fitted extent follows the new data, while a zoomed or panned range is kept. **Home** fits the plots, so they follow later arrivals again. Select latest changes only the selected record, not which ranges follow. A genuinely different coordinate, or a changed elapsed-time origin, resets main and My traces views and is explained in the status. With Select latest enabled, the selected mass spectrum advances to the newest acquisition record even if it sorts into the middle. Matrix, ion yield and an open My traces window refresh under current committed settings; peaks are not picked automatically. An earlier timestamp updates shared time zero, including unchecked sources, and the status explains the change. 3D views remain snapshots and indicate stale matrix data.

The settle delay is not a filesystem transaction. Changes are recognized by the index's modification time and size. Same-size edits with deliberately preserved timestamps, or made within one file-time step (about 1 ms on Windows), are therefore not seen; press **Stop**, then **Play** to reread the spectra. Export2D appends rows, so the index grows and a new measurement is always seen. Existing metadata/header changes trigger a full source reload; an incompatible grid leaves the accepted source intact.

The toolbar, minus, Play and Stop icons are drawn with the existing Matplotlib renderer, so they do not depend on special symbol fonts. The upward-swimming fish icon is embedded directly in the viewer and used for the main, folder-picker, Peaks, My traces and 3D windows. No separate icon file is required. The title uses the release date. Start this release with `python pim.py`; update any launcher that still refers to the previous viewer filename.

Each folder has an independent non-negative **Scale**. Typed scales update after one second; Enter applies immediately. Zero is allowed. Invalid text retains the last applied scale with a muted red field border and a hover explanation. Ring correction is calculated first, then optional acquisition-time normalization, then the folder multiplier. All mass channels share the same folder multiplier. These operations do not modify raw files or counts. Ion yields are separate curves for each included folder; identical y values remain independent points, without a connecting segment between them. Folder markers and dash patterns distinguish more than four sources. The lower-right ion-yield plot has no legend, keeping its data visible even when many folders are included.

The first loaded folder fixes the TOF grid. Every later folder must match every parsed bin centre exactly, including when the first folder is unchecked. A differing folder is rejected and never appears in the list. Even one readable spectrum with a different grid rejects the later folder. No tolerance, rounding, interpolation or resampling is used. Rejection leaves the committed peak and false-coincidence windows and subtraction setting unchanged. Ring factor, mass reference and integration windows are currently shared across included folders. Use separate workspaces for sessions requiring different correction factors or calibrations. Folder identity is not an inferred scan identity; per-scan controls are a later addition.

Settings appear in adjacent **Display** and **Integration** cards, always visible. The interface uses the dark palette and typography of i2PEPICO's `PreBD/prebd_plot.py`; spectra use the same cyan trace color. A compact three-row cockpit and one-line toolbar leave most of the main plotting column available for the graphs. Help is available on hover, without permanent message rows in Display or Integration. The navigation toolbar stays above the plots when the window is resized. Adding, removing, or including a folder fits the axes to the included collection and resets obsolete zoom history. Committed peak and false-coincidence bounds remain unchanged.

Each loaded folder has its own **filename mask** (the empty field shows an italic grey "Filename filter" hint) and a `matching/loaded rows` counter, for example `121/121 rows`. Enter one wildcard against the complete raw `.i2d` basename: `*120C*` or `*120c*.I2D` selects matching filenames, case-insensitively. A non-empty mask without `*` is tried as written first. If it finds no indexed filenames in that folder, PIMP retries with stars on both sides: `100` can match `my_100_file.i2d`, but selects only `100` if that basename is present. The entered mask is preserved; clearing it restores all rows. `?` matches one original Unicode code point; bracket classes are supported. Commas and semicolons are literal, not separators between masks. Paths and interior control characters are invalid. Leading/trailing whitespace is stripped. Blank restores every indexed row. Edits apply after one second or immediately on Enter. Pending or invalid text keeps the last applied mask and counter. Hover a clipped counter to see its full value. An unchecked folder still shows its mask count.

Filtering uses the already loaded measurements without rereading disk. Matrix, spectrum, ion yield, Peaks, My traces, numerical exports and new 3D views use the rows retained by included folders' applied masks. Exports contain every retained row, on the full TOF grid, whatever the zoom, and list each row's folder, original row, file, timestamp and acquisition time; they also record each folder's applied mask. Existing 3D snapshots keep their data and become stale. Full loaded folders still determine TOF compatibility and the time origin. If a mask removes every usable value of the chosen y coordinate, PIMP falls back to another usable coordinate and reports that change in the status line. A subset with no usable display coordinate is rejected; clear or edit the mask to recover. A mask matching no filenames is accepted and clears the plots. Clearing it restores data. Monitoring continues loading complete folders: matching arrivals increase both counts; other arrivals increase only the denominator. **Select latest** selects only retained new rows, preserving an existing valid selection

when none matches.

Workspaces store both the applied mask and draft text. Manual save and data actions apply usable drafts first; autosave preserves drafts without applying them. Restored drafts remain pending until edited, applied with Enter, or flushed by an explicit save/export action. PIMP reads version 1 workspaces with blank masks and version 2 workspaces with the My traces defaults, and writes version 3; older viewers reject version 3. Zero-match workspaces can be saved, reopened and cleared.

Select the desired **y-axis** parameter. Common choices are:

- `E` for a photon-energy scan
- `Watlow` for a temperature scan
- `MFC10`, `MFC11`, or `MFC12` for a flow scan
- `P_roughing` or `P_foreline` for pressure-dependent data
- `TimeOnStream_h` for time-on-stream curves
- `MeasTime` (**Integration time [s]**) to inspect acquisition durations

The viewer retains the absolute `MeasStart` timestamps written in each index; these, not file modification times, supply time on stream. Time on stream is measured from the earliest valid timestamp among all loaded folders, including unchecked folders and rows hidden by file masks, so inclusion toggles and masks retain the same origin. Time-sliced folders have no wall-clock timestamps and never contribute an origin; their TDC time is a separate axis (see Time-sliced folders). Adding or removing the earliest folder can change that origin. The axes name the actual zero instant; the status reports a change to that instant. Unknown timestamps remain unavailable on this axis. Multiple folders without absolute timestamps are not aligned by assuming that their local time zeros match. Current Export2D indices provide channel labels and units in their `Parameter:` lines. For older indices without those lines, the viewer uses built-in labels where available and otherwise displays the short column name. Duplicate column names are ambiguous and must be re-exported with Export2D.

The first folder selects its most frequent exact TOF grid, with input order breaking ties. A spectrum in that folder on a different grid is flagged as incompatible and omitted rather than resampled. For later folders, any such difference rejects the entire addition. The status reports nonzero absent, unreadable and incompatible spectrum counts.

The upper plot shows the PI or TPI matrix. Click a row to display its mass spectrum in the lower-left plot. Equal energies, temperatures, or other y values occupy the same band and overlap. Click again to cycle the measurements at that coordinate; the selected row comes to the front, and its source number/file identifies it below. Source numbers follow the folder list. Measurements are never combined at equal coordinates. A row missing its selected y coordinate remains in the analysis and export but cannot be placed in the matrix; the status reports the number of hidden rows. Drag horizontally over a TOF range to obtain the ion yield in the lower-right plot. A drag changes the integration window and retains the selected measurement.

## Time-sliced folders

`Export2DSlices`, `Export2DSlicesWindow` and `Export2DSlicesTrigger` (see `Export2DSlicesExample.scr`) split **one closed raw file** into time slices and write one index row and three TOF spectra per slice, named `<raw file>_s00000.i2d`, `_s00001.i2d`, and so on. The index contains one extra line, which PIMP uses to recognize such a folder:

- `Slicing: Time <width_ms>`: absolute TDC time from the start of the file;
- `Slicing: Trigger Ch<n> <width_ms> <max_delay_ms>`: delay after the latest hit on trigger channel n (1 to 16).

**Windowed folders.** `Export2DSlicesWindow` exports only the full slices between a start and an end time. The slices stay on the whole-file grid and keep their absolute times, so the slice axis starts at the window's first slice, at its real time in the file, not at 0. File numbering restarts at `_s00000` in each window; PIMP takes each slice's position from its `MeasStart`, not from its name. The extra line `Window: requested <start> <end>` records the request and is ignored by PIMP: the exported range is given by the rows, from the first slice's start

to the last slice's end. Windowed and whole-file exports of the same raw file therefore share one time axis. Each window goes into its own folder.

An index without a `Slicing:` line is an ordinary export and behaves exactly as before. In a sliced index, `MeasStart` is the slice midpoint in ms, not a date. It becomes the y-axis **Time [ms]** (`SliceTime_ms`) or **Delay after Ch*n* [ms]** (`TriggerDelay_ms`), always in ms, and is the default y-axis when the folder is opened. Constant parameters such as `E` remain selectable. A sliced folder has no wall-clock timestamps, no time on stream and no time zero.

Rules:

- **Separate folder, opened offline.** Export into an empty folder that is not named `PIMP`. A sliced index in a folder named `PIMP` is rejected, including by Play/Stop monitoring, which keeps retrying as for any unavailable input. Sliced folders are never monitored.
- **Let the export finish before opening the folder.** Every listed slice needs readable center, ring and all spectra on one TOF grid; an index without rows, a missing or unreadable spectrum, or an index that changes while its spectra load is rejected with advice to reopen later. PIMP reads only the rows the index lists. A stable index is not proof that the raw run was completely exported: a partly written export cannot be distinguished from a shorter one.
- **One folder only.** A time-sliced folder must be the only loaded folder, including unchecked, masked and waiting folders. Adding it next to other folders, or adding anything next to it, reports *A time-sliced folder must be the only loaded folder; remove the other folders first.* In a group, folders are decided in order: [ordinary, sliced] adds the ordinary folder, [sliced, ordinary] adds the sliced one. Workspaces that combine a sliced folder with other folders, waiting folders or a saved watch folder are rejected before anything changes.
- **0 to 24 hours.** Every listed slice, including its full width, must lie within 0 to 86,400,000 ms of time or delay. A folder with a slice beyond 24 hours is rejected, never truncated or rescaled. This limits TDC time or trigger delay, not the summed exposure in seconds.
- **Slice width: a whole multiple of 0.01 ms, at least 0.02 ms.** Export2DSlices rounds the width you give to the nearest multiple of 0.01 ms (10 µs) before slicing, and the `Slicing:` line records that rounded width; 0.015 ms becomes 0.01 or 0.02 ms, and a width that rounds below 0.02 ms stops the export with a message. PIMP rejects any other width. On this grid the TDC slices exactly the recorded width and absolute-slice exposures print exactly. 0.02 ms is recommended: even multiples of 0.01 ms also give exactly printed midpoints; odd multiples such as 0.03 ms give midpoints ending in 5 µs, which the index rounds to two decimals. Each printed midpoint must be (k + ½) × width within that 0.005 ms print rounding, and each slice k may appear once.

Display: with the slice axis selected, each slice is drawn exactly from k × width to (k + 1) × width, so a single slice has its real width. Slices omitted by the export or hidden by a file mask leave an empty gap, and ion-yield and breakdown lines break across the gap. The gap is display only: it never enters numbers, integrations, breakdown results, exports or 3D views. Selected-row labels show two decimals, e.g. `Delay after Ch3 = 0.03 ms`. With `E` or another parameter selected, the ordinary geometry applies.

`MeasTime` stays in seconds: the slice width for absolute slicing, or the exposure of that delay bin summed over trigger periods. **Counts / s** divides by it as for any folder, so a delay bin that no trigger period reached (exposure 0) shows as missing (NaN) only with counts per second. i2PEPICO prints `MeasTime` with five decimals, i.e. to 10 µs. Absolute-slice exposures are whole multiples of 10 µs and therefore exact. Summed trigger exposures are rounded to 10 µs: the relative error is small for well-filled bins, but a delay bin with less than 5 µs of exposure is written as 0 and shows as missing under counts/s; counts are not affected.

The folder's hover help shows the raw `Source:` file. Numerical exports add `slicing=<the Slicing line>` and `slice_source=<the Source path>` to their metadata, also when `E` is selected; the y column carries the printed ms values under `SliceTime_ms` or `TriggerDelay_ms`, and the `timestamp` column is `NaT`. Workspaces store no new fields and restore the saved y-axis strictly: if the folder was later re-exported so that the saved axis no longer exists, the workspace is rejected; open the folder afresh.

**Tested size.** Sliced folders are tested up to 1,000 slices. One measured load of a synthetic folder with 1,000 slices × 5,000 TOF bins (3,000 spectra, covering exactly 0 to 24 h) took 2.4 s on a Windows 11 laptop (Intel Core Ultra 7 265H, 64 GB RAM, Python 3.13.14, NumPy 2.5.3) with the files most likely in the operating-system cache. The process working set afterwards was 174 MB, and its peak working set, including Python start-up and reading, was 576 MB. The three count matrices alone occupy 120 MB. Larger numbers of slices are not forbidden but are untested; memory use and loading time grow in proportion to the number of slices.

## Available views

- **TPI:** threshold photoionization, calculated as `Center - f × Ring`
- **PI (all e-):** ions coincident with all detected electrons
- **Center (raw):** the uncorrected central electron region
- **Ring (reference):** the ring contribution used for subtraction

Use **Ring factor f** in **Display** to adjust the ring subtraction. Typed values update after one second; Enter or the spin buttons apply them immediately. The factor is displayed with three decimals and spin steps of 0.001. Typed factors with more than three decimals are marked invalid rather than silently rounded; the last applied factor stays active. A changed value is amber, labelled Pending in its hover text, until the one-second delay or Enter applies it. A rejected value is a muted red and stays visible while the last applied value continues to be used. Focus does not hide either cue. Workspaces use the same three-decimal validation. Restored text that differs from the applied value is shown as pending or invalid and is not applied merely by opening the workspace. The factor is enabled for the TPI view. **Counts / s** divides each spectrum by `MeasTime` (integration time in seconds).

For photon-energy scans, the TPI ion yield is labelled **ms-TPES** for a selected mass range, or **TPES (all masses)** for the full mass range. The PI ion yield is labelled **Photoionization spectrum**. Temperature, time, and other scans use ion-yield labels.

The **Display** card has three rows: **View**, **Ring f =** and **Intensity** (Linear/Log and 1-99%/Full range); **x-axis** (TOF / µs or m/z) and **Calib** (µs to m/z); then **y-axis**, a separator, **z-axis: Counts / s**, another separator and **Clear selection**. Peaks and My traces open from the toolbar. Enter a known calibration pair, TOF in µs and its corresponding m/z. Both reference values are validated together one second after the last edit; Enter applies them immediately. Hover over either reference field for the one-point formula. There is no separate Calibration tab or Apply button.

Changing the reference while displaying m/z rebuilds the matrix and fits the new mass axis, clearing history in the previous calibration. In TOF mode, the matrix and its zoom are retained. Incomplete or invalid reference text leaves the last valid reference active. Pending edits use an amber border and invalid edits use a muted red border; hover over them for the applied values and the reason. The reference pair is accepted or rejected together. No popup interrupts typing or dragging.

## Integration and false-coincidence subtraction

Scroll the mouse wheel over the matrix to zoom in or out around the pointer. Hold Shift for horizontal-only zoom, or Ctrl for vertical-only zoom. With both held, zoom both axes. Mouse modifiers work even while a numeric entry has keyboard focus. Use the magnifying glass or pan tool in the toolbar for rectangular zoom or panning. Click that same active tool again to resume selecting spectra and integration windows. **Home** fits the current matrix, mass spectrum, and ion yield, including any changed integration or subtraction settings. Back and Forward navigate wheel zoom with one history step per wheel event. Changing the scan axis, TOF/mass units, calibration, or a manually replaced dataset resets navigation history. Automatic watch refresh keeps navigation history and manually zoomed ranges; fitted ranges follow new data.

**Zoom rescales** is enabled by default: color limits are recalculated from the bins visible in the matrix after zooming, panning, or navigating history. Turn it off to keep the full-matrix intensity scale for comparisons. The **Intensity** Linear/Log choice applies to matrix color intensity, mass-spectrum intensity, ion-yield intensity, and 3D height. The adjacent colour-range selector offers **1-99%** (startup default) and **Full range**, beside Linear/Log. Full range uses the complete finite intensity range. 1-99% uses actual finite 1st and 99th

percentiles in the same region selected by Zoom rescales, saturating values outside the bounds. Positive Log uses positive values; signed Log uses symmetric bounds from the percentile magnitudes and the existing relative threshold. Collapsed/constant/sparse percentile bounds fall back to Full range. A view with no finite data retains the previous color limits. Signed intensities use a symmetric logarithmic scale in Log mode. Nonpositive values in a positive-only log display are masked; their raw values and numerical exports are retained. Linear preserves intensity ratios, so weak channels may be hard to see beside a very strong peak. Log is useful for finding those weak channels, but it also makes background noise more visible. Neither option changes measured counts, integration or numerical exports. The same explanation is available by hovering over the intensity selector. Colour contrast changes neither scientific values nor 1D intensity-axis limits. A 3D snapshot keeps its captured colour limits.

The **Integration** card has vertically aligned **Peak** and **False coincidence** radio buttons beside their respective TOF bounds. Select a row to choose the drag target, then drag across the matrix or enter numerical bounds. The fields follow the main x-axis: **TOF / µs** or **m/z**. Stored integration limits always remain TOF. Recalibration updates their mass display without moving the selected bins. Valid pending mass edits are interpreted with the previous reference before recalibration; usable endpoints in an unfinished draft are converted, while unfinished text and the previous applied selection remain. Typed bounds update **1 second after the last edit**; dragging updates the fields immediately without scheduling another update. Incomplete, reversed, nonfinite, or empty selections keep the last valid result. Pending bounds are amber and rejected bounds are muted red; hover for the applied window. An empty false-coincidence field that has never been used stays in the ordinary style. An invalid draft does not replace the applied window.

Choose **False coincidence** and drag over a representative background. This enables and ticks **Subtract false coincidences**, which is greyed out until a false-coincidence window is applied; afterwards you can untick and tick it freely. Peak and reference windows are independent: a reference entered as 12.5–13 µs stays at those bounds, including when the peak covers the whole TOF range. No overlap conflict check is imposed. The peak and reference are marked in orange and red, respectively. A numerical integration update retains the matrix zoom. A drag replaces unfinished text for its own window; unfinished input in the other window never prevents the drag.

The selected false-coincidence intensity is scaled by the ratio of included TOF bins and subtracted from the ion yield. It also applies independently to each My traces channel. An applied false-coincidence window containing no measured bins (possible after adding folders with a different TOF range) makes corrected yields undefined in both paths; the ion-yield title says so, and the ion yield and My traces exports are refused with an explanation. A workspace saved by an earlier release with subtraction on but no false-coincidence window opens with it off, and the status line says so. Subtraction does not alter the displayed matrix, mass-spectrum preview or their exports. Peak search applies the row-wise constant subtraction described under Peak definitions, and refuses a missing or unsampled false-coincidence window. Ring subtraction with f is a separate operation on the TPI matrix and mass spectra.

Mass calibration transforms the original TOF positions and bin edges into non-equidistant mass channels. Values are counts per original channel, not a density: no Jacobian or channel-width correction is applied. Switching units or recalibrating does not change counts or the sum over a stored integration range.

## Saving and restoring an analysis

PIMP autosaves the current workspace beside `pim.py` as `pim.autosave.pepico.json` every ten seconds when it has changed and once more on exit. Autosaving records unfinished field/peak-editor text alongside committed values without applying those edits. On startup, PIMP first checks that the saved folders are still there, without loading them; the status bar says so while it waits, for example for a network drive. When every folder is there, or when only peak definitions were saved, it asks whether to reopen the workspace. When some folders are missing, it lists them and asks whether to reopen the workspace without them; they are dropped from the autosaved workspace when it is next saved, so closing PIMP at once keeps them for the next start (for example after reconnecting a drive). If time on stream is shown, it then starts at the earliest remaining folder. When none of the folders is there, PIMP asks nothing and the status bar says why. The committed TOF/mass calibration, the export file type, the export folder and the **+** folder are saved in workspaces and kept at startup whatever happens to the workspace: when you decline, when folders are

missing, or when source files can no longer be loaded. Declining a restore does not erase the previous workspace until the new session changes. Monitoring stays stopped when reopening. An unwritable program directory produces an inline autosave explanation; manual Save workspace remains available.

Choose **Save workspace** to write a version 3 `.pepico.json` file. It saves all loaded folders, inclusion flags, scales, every analysis control, displayed measurement preview, three plot viewports, active navigation tool and window geometry. It preserves both unfinished numeric text and the last valid applied values. Valid pending edits are applied before saving. Unfinished text is restored with marked fields and hover explanations, without starting a timer. Opening Peaks retains the existing picker validation: valid restored thresholds apply immediately; invalid thresholds keep their applied values. The existing TPI ring-factor export flush and open-picker flush can also apply valid restored text. Other paused drafts apply only through their existing explicit commit/flush paths. It also saves the complete peak list, editable indices/labels, channel selection, picker settings, peak-editor units and unfinished text, My traces folder selections, trace visibility independently of inclusion, sum visibility, the fixed fraction range, Fractions/Absolute mode, the channel legend, window geometry, My traces view limits and navigation tool. Peaks and My traces windows that were open are reopened automatically. Their settings remain available when they are closed. The existing 3D view remains a temporary snapshot. The configured watch path, five/ten-second poll interval, Select latest preference and colour range are saved. Monitoring restores stopped; press a folder's Play button explicitly. Waiting folders are saved as pending cards; they contain no loaded spectra. Version 1 files migrate with blank masks, and version 2 files open in Fractions mode without a channel legend; version 3 is written. Future versions and incomplete files are rejected before the open analysis changes.

**Open workspace** replaces the current analysis after validation, reloads the original folders and then restores their settings. The file contains paths and settings rather than copies of spectra. Save the current workspace first to keep it. Keep the export folders available. Missing sources, malformed settings, or any incompatible grids leave the current analysis intact and show the problem. Every saved folder is required, including excluded ones, except waiting PIMP folders; all grids must match the first saved folder exactly. Incompatible or unsupported files are rejected even if a differing folder is unchecked. Opening a workspace from another computer keeps your export and **+** folders when its folders do not exist here. Saving uses an atomic file replacement to avoid truncated workspace files.

Paths are relative to the workspace directory where possible; the remembered export and **+** folders are stored as absolute paths. Move the workspace and export folders together while retaining their relative structure to open the same analysis elsewhere. Relative paths use forward slashes on Windows, Linux and macOS; absolute Windows drive paths require adjustment on another platform. Navigation history is recreated at the restored view; its full stack and 3D preview windows are not stored.

## Optional 3D view

**3D view** opens a separate rotatable point-and-stem plot of the visible matrix region. Each measured bin is a point at its actual TOF/mass, scan coordinate and intensity. Equal coordinates remain independent measurements. The stationary view includes original finite bins supported by the intensity scale, without pooling, averaging, or interpolation. Positive-only Log omits nonpositive points from the view; their numerical counts remain unchanged. Translucent vertical stems make narrow TOF peaks easier to follow. To avoid millions of line segments, stems use each measurement's original minimum and maximum bins in 24 TOF intervals, plus endpoints; all stationary points remain. Colors and the colorbar use the exact same colormap and frozen intensity limits as the matrix at opening; depth shading is disabled to preserve those colors. Linear/Log height follows the main intensity setting, with tick labels in counts or counts/s rather than transformed units. False-coincidence subtraction is not applied to the 3D view.

Drag to rotate. During rotation, panning or zooming, **Moving preview** temporarily shows stems from the original minimum and maximum bins within 8 TOF intervals per measurement, plus both endpoints. This retains narrow peaks and negative features while using at most 18 original bins per row. No counts are averaged and every measurement row remains independent. On release, mouse departure, or loss of window focus, it restores full quality and keeps the camera angles. Re-entering with a mouse button still held resumes the moving preview without needing a new click. There is no row inspector or manual quick-preview control. **Full quality** identifies the stationary view. Toolbar image saving always restores full quality before writing.

Stems reach zero in Linear or signed Log; positive-only Log uses its minimum displayed intensity as the base, since zero cannot be represented on a logarithmic axis. This base is labelled.

The window is a timestamped snapshot. Main-viewer edits mark it **Viewer changed**; reopen it to see current results. Changing the main zoom does not change an already open 3D snapshot. Camera angles and these windows are not saved in a workspace.

## Exporting results

**Menu.** The **Export** menu (toolbar button or Alt+A) uses the names the window shows: the matrix and ion-yield entries carry their panel titles, and Peaks and My traces the names of their toolbar buttons. The entries above the separator export measured data, the two below it your analysis:

- **TPI matrix...**, or **PI matrix...**, **Uncorrected TPI matrix...** or **Ring-reference matrix...** for the other views, exports every retained measurement of the included folders on the full TOF grid, whatever the zoom. TPI includes ring subtraction, and folder multipliers are applied. False-coincidence subtraction is not applied. Colour clipping does not change the values.
- **Mass spectrum...** exports the measurement shown in the mass-spectrum panel on the full TOF grid, with the same matrix processing and no false-coincidence subtraction.
- The ion-yield entry is named like the ion-yield panel, for example **ms-TPES...**, **TPES (all masses)...** or **Photoion yield...** (**Ion yield...** without data). It exports the sum over the applied Peak window for every retained measurement: one table for all included folders, with a `folder` column telling which trace each row belongs to. With **Subtract false coincidences** ticked, the false-coincidence correction is included. A false-coincidence window, or a Peak window, without measured bins on the current TOF grid would make every yield undefined, so the export is refused with an explanation.
- **Peaks...** exports the peak definitions: windows, not intensities. A new peak that has not been added is not a definition. It applies a pending calibration edit only; other pending fields do not affect peak definitions.
- **My traces...** exports the rows of the folders selected in My traces, with channel yields, their sum and fractions. Channels that are included but hidden stay in the sum and in the file. With subtraction ticked, a false-coincidence window without measured bins refuses the export with an explanation; so do no included channel and no selected folder.

**What happens when you export.** Pending valid edits (folder scales and masks, ring factor, calibration, integration windows and the Peaks editor) are applied first, also when you then cancel the save dialog, exactly as Enter would apply them. Text that cannot be applied keeps the last valid values; it does not block the export, and the file lists the field. While the save dialog is open, the status bar shows what will be exported. The dialog starts in the folder of your last export, remembered across sessions; otherwise in the folder that contains `pim.py`, or in your home folder if that folder is read-only. It suggests a name made of the first included folder's name (with `+1`, `+2`... for further included folders), the view and the artifact, for example `DMF_2023-09-18_fine_v1+1_tpi_ion_yield_mz72p12-74p32.csv`. **The extension of the file name decides the format**: `.csv` writes comma-separated values, `.txt` (or `.tsv`) tab-separated text. The file-type list in the dialog may not change the extension on every system, so to get tab-separated text make sure the name ends in `.txt`. The last choice is suggested next time. An existing file is replaced only after the new one is completely written, so a failed export leaves it unchanged and the error message says so. On Windows this happens, for example, when the file is open in Excel: close it and export again. After a successful export the status bar names the saved file, its number of rows and its folder.

**File layout.** Every export is UTF-8 text. It starts with metadata lines of the form `# key=value`, one per line; a value is everything after the first `=`. Line 3 is `# table_header_line=<n>`, the line number of the first header row, and line 4 is `# header_rows=<k>`, the number of header rows. The header rows are not commented; the first holds the column names. Data rows follow: one per measurement (matrix, ion yield, My traces), per TOF bin (mass spectrum) or per peak (Peaks). Text that contains the delimiter, a quote or a line break is quoted as in any CSV file.

- Matrix: the scan coordinate, named as in the index (for example `E`), then one column per TOF bin. It has two header rows whatever the display axis shows: the TOF bin centres in µs, and a row starting with `mz` that

holds the m/z of each bin (nan below zero TOF). Intensities are counts per bin in both cases.

- Mass spectrum: tof_us, mz and intensity.
- Ion yield: the scan coordinate and integrated_intensity.
- My traces: the scan coordinate, sum_yield, then one yield[...] and one fraction[...] column per included channel, named by the channel title.
- Peaks: peak_id, index_mz, label, low_us, high_us, low_mz, high_mz, included, visible, method, note, origin, overlap_indices (a JSON list), centre_us and sigma_us (empty for a manual peak).

Matrix, ion yield and My traces tables end with six columns that identify every row: folder (the folder's name in the folder list), source_id, source_row (the row in that folder's index; zero-based), file, timestamp and acquisition_time_s. Rows are in the matrix order, so with several folders the rows of different folders interleave by coordinate; equal coordinates stay separate rows. To plot one trace per folder, as the ion-yield panel does, group or filter by folder. If the scan coordinate has the name of another column of the table, its column is called <name>_y. Axes, coordinates and window bounds are written exactly (they read back to the same number); intensities, yields and fractions with ten significant digits. Undefined values are nan. Input that could not be applied is listed in unapplied_input lines, and warnings in warning lines.

**Key reference.** The common keys, in file order (M matrix, S mass spectrum, Y ion yield, P Peaks, T My traces):

| key | in | value |
|---|---|---|
| (PIMP export) | all | first line |
| format | all | 2 |
| table_header_line | all | line number of the first header row |
| header_rows | all | number of header rows: 2 for the matrix, otherwise 1 |
| artifact | all | matrix, mass_spectrum, ion_yield, peaks or my_traces |
| summary | all | the sentence shown in the status bar |
| pimp_release | all | release date of pim.py |
| exported_at | all | local date and time of the export |
| view | M S Y T | the View choice, for example TPI |
| matrix_calculation | M S Y T | center-f*ring, all, center or ring |
| ring_subtraction_factor | M S Y T | f; in TPI view only |
| intensity_unit | M S Y T | counts or counts_per_second |
| intensity_order | M S Y T | order of ring correction, normalization, folder multiplier |
| mass_calibration | all | one_point_zero_offset (formula below) |
| reference_tof_us | all | calibration TOF in µs |
| reference_mass_mz | all | calibration m/z |
| y_column | M S Y T | name of the scan-coordinate column |
| y_label | M S Y T | y-axis label shown in the window |
| elapsed_time_origin | M S Y T | zero of time on stream, or None |
| slicing, slice_source | M S Y T | time-sliced folders only |
| source | all | one JSON object per loaded folder (repeated) |
| (artifact keys) | | listed below |
| unapplied_input | all | one field that could not be applied (repeated) |
| exported_values | all | last_valid_applied_settings, with unapplied_input |
| warning | all | one warning (repeated) |

The calibration is mz = reference_mass_mz * (tof_us / reference_tof_us)^2. A source object holds id, label (the name in the folder list), folder, index_name, included, scale (the folder multiplier) and file_mask; in My traces also in_my_traces. The artifact keys are:

- Matrix: rows, tof_bins, false_subtraction=not_applied_to_this_artifact, columns (describes both header rows), and a warning when bins lie below zero TOF.
- Mass spectrum: measurement_file, measurement_folder (its path), measurement_folder_label, measurement_source_id, measurement_source_row, measurement_timestamp, measurement_y (its scan coordinate), acquisition_time_s, tof_bins and false_subtraction=not_applied_to_this_artifact.
- Ion yield: rows, spectrum_title, peak_window_tof_us and peak_window_mz (low,high; m/z is nan below zero TOF), peak_bins, false_subtraction (off or applied); when applied also false_window_tof_us, false_window_mz, false_bins and false_scale (Peak bins divided by false-coincidence bins); last calculation, peak_sum or peak_sum-false_sum*false_scale. A warning counts rows whose yield is

nan.

- Peaks: peaks, the number of definitions.
- My traces: rows, my_traces_excluded_sources (JSON list of the folder IDs not selected in My traces), one channel JSON object per included channel (peak_id, title, index_mz, label, low_us, high_us, low_mz, high_mz, visible, method, centre_us, sigma_us), false_subtraction; when applied also false_window_tof_us, false_window_mz, false_bins and false_correction=channel_sampled_bins/reference_sampled_bins (each channel is scaled by its own sampled bins); then fraction_denominator (fractions use the sum of included channels before the folder multiplier) and hidden_channels_in_denominator=True. The My traces warnings follow.

**Reading the files.** These Python recipes were run on exports of all five artifacts, comma- and tab-separated. First read the layout:

```
from pathlib import Path

def read_layout(path):
    """Delimiter, header line, header rows and metadata of a PIMP export."""
    metadata = []
    with open(path, encoding="utf-8") as stream:
        for line in stream:
            if not line.startswith("# "):
                break
            key, _, value = line[2:].rstrip("\n").partition("=")
            metadata.append((key, value))
    delimiter = "\t" if Path(path).suffix.lower() in (".txt", ".tsv") else ","
    return delimiter, int(metadata[2][1]), int(metadata[3][1]), metadata

d, header_line, header_rows, metadata = read_layout(path)
```

pandas, for a table with one header row:

```
import pandas as pd

table = pd.read_csv(path, sep=d, skiprows=header_line - 1, engine="python",
                    keep_default_na=False, na_values=["nan"])
```

engine="python" is needed because the faster engine can fail on a quote in a skipped metadata line. The NA options keep a folder or file named NA as text, while nan stays a missing value; empty cells, such as centre_us of a manual peak, are read as empty text. Do not use comment="#": it also cuts data fields that contain #. For the matrix, read both header rows as TOF and m/z column levels, or skip the m/z row:

```
matrix = pd.read_csv(path, sep=d, skiprows=header_line - 1, header=[0, 1],
                     engine="python", keep_default_na=False, na_values=["nan"])
matrix = pd.read_csv(path, sep=d, skiprows=list(range(header_line - 1)) + [header_line],
                     engine="python", keep_default_na=False, na_values=["nan"])
```

NumPy, matrix axes and values:

```
import numpy as np

tof_bins = int(dict(metadata)["tof_bins"])
tof, mz = np.loadtxt(path, delimiter=d, quotechar='"', skiprows=header_line - 1,
                     max_rows=2, usecols=range(1, 1 + tof_bins), comments=None)
values = np.loadtxt(path, delimiter=d, quotechar='"', skiprows=header_line - 1 + header_rows,
                    usecols=range(0, 1 + tof_bins), comments=None, ndmin=2)
y, intensities = values[:, 0], values[:, 1:]
```

quotechar is needed because a provenance field may contain a quoted line break, and ndmin=2 keeps a one-measurement file two-dimensional.

Origin: import the file with table_header_line - 1 header lines (the metadata); the next line holds the long names. For the matrix, use the m/z row as a sub-header line, or skip it.

Excel: Data > From Text/CSV can import a single column, because the first line of the file, a metadata line, has only one field. Instead choose Data > Get Data > From Other Sources > Blank Query, then Home > Advanced Editor, paste the query below, set Path to the file and Delimiter to ",", or "#(tab)" for a .txt file, and choose Close & Load. The query reads lines 3 and 4 itself, skips the metadata, takes the column names from

the first header row and leaves out the matrix's m/z row. It reads numbers with a decimal point whatever the Windows number format; `nan`, `inf`, other text and empty cells stay as they are.

```
let
    Path = "C:\Data\exports\example.csv",
    Delimiter = ",",
    Lines = Text.Split(Text.FromBinary(File.Contents(Path), 65001), "#(lf)"),
    HeaderLine = Number.From(Text.AfterDelimiter(Lines{2}, "=")),
    HeaderRows = Number.From(Text.AfterDelimiter(Lines{3}, "=")),
    Table = Csv.Document(Text.Combine(List.Skip(Lines, HeaderLine - 1), "#(lf)"),
        [Delimiter = Delimiter, QuoteStyle = QuoteStyle.Csv]),
    Named = Table.PromoteHeaders(Table, [PromoteAllScalars = true]),
    Data = Table.Skip(Named, HeaderRows - 1),
    Numbers = Table.TransformColumns(Data, {}, each try Number.FromText(_, "en-US") otherwise _)
in
    Numbers
```

**Format change.** Files written before this release (with commented column names, `row_provenance=` metadata and, for the matrix, the TOF axis only in a comment) are not converted. Scripts that read exports with `np.loadtxt(path)` need the recipes above.

## Peak definitions

First check the main one-point mass reference; automatic integer peak indices use that calibration, then choose **Peaks**. **Pick peaks** uses an explicitly selected measurement (a matrix click or a new Select latest arrival). The automatically displayed initial spectrum is a preview, not a selection. **Clear selection** removes the selection marker without removing data. Without an explicit selection, picking sums all measurements of included sources on the original TOF grid; it does not average or restrict rows to the visible matrix range. Sorting and monitoring preserve explicit selection by source and original row; excluding, removing or losing that measurement clears it. Workspaces retain their existing preview, but do not persist the new explicit-selection state.

Both inputs use the current PI/TPI/etc. view, applied ring factor, Counts/s setting and source multipliers, using linear numerical values even with Log display. When false-coincidence subtraction is enabled, each contributing row has its constant background subtracted before summing: the finite-value sum inside the applied false-coincidence window divided by its total sampled-bin count. Missing samples count in that denominator. Negative values and missing-data gaps are preserved; a bin without valid contributors remains unavailable. A missing or unsampled false-coincidence window refuses picking. A selected row without usable false-window data refuses; the sum instead omits such rows, reports their count and refuses if none remain. The picker labels its input **Selected measurement: ...** or **Sum of N measurements**, including subtraction and any omissions. **Visible spectrum only** then restricts this input to the zoomed lower-left TOF range (converting m/z limits to TOF). Noise × σ and minimum relative prominence control detection sensitivity. Threshold edits validate after one second or Enter. Each valid threshold applies independently; an invalid threshold retains its last applied value and marks only its own field. **Visible spectrum only** applies independently of threshold text. Invalid editor drafts or thresholds do not block picking with applied settings, showing a stored peak in the main plot, exports, saving, or closing Peaks.

The picker estimates baseline-subtracted left/right half-height crossings and fits a Gaussian core when enough original bins constrain it. The proposed range is centre ±2.575829σ, the central 99% range of that Gaussian. Unsuitable cores use a marked FWHM estimate; boundary/unresolved peaks use marked measured support. This is model coverage, not a confidence interval or a promise that 99% of an asymmetric or blended measured peak lies inside. Review the Method and Review columns, and adjust bounds when needed.

The table shows each interval in both µs and calibrated m/z. Each channel has a unique integer **Nominal m/z (integer)** identifier and an optional label, such as `C0 loss`. That identifier does not move the integration window. Calibrated low and high bounds may be fractional. Fractional text in the identifier is rejected and the last applied integer remains. Picking an integer already in the list expands its TOF interval by union and keeps its label, identity and channel selection; it never shrinks an existing manual range. Multiple detections with the same integer form that one channel.

The editor row above the table is either a new draft or the selected peak. With no peak selected it is a draft: type its fields, or drag a Peak range on the main matrix while Peaks is open, which fills Low/High and a

nominal m/z from the range centre; then choose **Add peak**. A drag always starts a new draft: it leaves the selected peak (committing a valid unapplied edit) and never changes a stored definition. To refine a stored peak, type its bounds. A draft whose integer already exists is not added. With a peak selected, the editor edits it; valid changes apply after one second or Enter. **Escape** reverts an unapplied edit; with nothing to revert, it leaves the selected peak and empties the editor. A click on the empty table space below the rows does the same after committing a valid edit. Delete selected supports several selected rows; Clear asks before removing the entire definition list. These actions never change raw data.

Use the editor's **TOF / µs** or **m/z** units. Switching units converts each usable draft endpoint and preserves unfinished/invalid endpoint text; it never requires the whole draft to be valid. Adding or applying a definition still requires a unique non-negative integer index and finite ascending non-negative bounds. Stored bounds always remain TOF; recalibration updates displayed mass bounds while preserving TOF and the editable integer identifier. If calibration changes during an unfinished mass-bound edit, the draft is warned and the last applied definition is retained. Incomplete or duplicate integer edits are marked rather than applied. Selecting another row first commits a valid edit to the previous definition, including within the one-second delay. Invalid text never blocks choosing a peak: selecting a row replaces the draft with that peak's valid stored values, using the current calibration. Clicking the same row or pressing Enter in the table reloads it too. Selecting a row discards an unadded new draft without adding it. **Escape** also restores the current stored definition, or clears an unadded draft. These rules apply after reopening Peaks or restoring a workspace. Automatic display refresh preserves unfinished text until an explicit choice. New drafts are only added by **Add peak**. Escape cancels any pending editor timer; it does not remove stored peaks or change calibration. Refreshing spectra/corrections keeps list positions.

Overlapping rows are coloured and name the conflicting indices. Inclusive touching boundaries also count as overlaps. Warnings update after every change; ranges are never silently trimmed or merged across different integer channels. The selected definition is highlighted on the main mass spectrum. Definitions remain available after their original source folder is removed. Selecting a single row, by a click, Enter or the Up/Down keys (once the key is released), makes its integration bounds the main Peak range and plots its ion yield; it replaces a range dragged or typed in the main window, including unapplied text. A peak without measured TOF bins leaves the main range unchanged. Selecting several rows, scrolling, or a refresh of the list does not change the main range. Main bounds and the matrix span use the current TOF/mass units, with canonical TOF preserved. Selection leaves the Peak/False coincidence choice, the matrix zoom and the stored false-coincidence range unchanged.

## My traces

Choose **My traces** in the toolbar. This window shows the selected channels' traces and their sum along the scan. It uses the main matrix view and applied ring factor, normalization and false-coincidence window. TPI, PI (all e-), raw Center and Ring can all be analysed. Its scan x coordinate follows the main y-axis, so temperature/time/flow scans are supported too.

**Fractions** plots the sum on the left axis and the visible channels' fractions on the right axis. **Absolute** plots the sum and the visible channels' corrected yields on one axis, in counts or counts/s; the fractional axis and **Fix fractions** are hidden. Switching mode works like Home in the new mode and restarts Back/Forward history; clicking the selected mode again changes nothing. In Absolute mode the intensity axis is fitted to the whole of the shown traces by Home and whenever a trace is shown or hidden, even after a manual zoom; the scan range is kept. In Fractions mode showing or hiding the sum refits the sum axis the same way; other Show changes keep the view. Zooming and panning otherwise keep the intensity range.

The title follows the main view and y-axis: fractional TPI data against photon energy form the **Breakdown diagram**; other fractional plots are titled **Fractional** and the main ion-yield label (for example **Fractional photoionization spectrum**), and Absolute plots use the main ion-yield label itself (for example **ms-TPES**).

Each channel has separate **Include** and **Show** checkboxes. **Include** controls the summed spectrum and fraction denominator, matching PreBD. **Show** hides only the channel's trace. The **All** row below the column titles includes, shows or hides every channel at once; its boxes are partly filled when only some channels are selected, and a click then selects all. Show covers the **Sum** row and the included channels only. The **Sum** row's Show box shows or hides the sum. Hidden included channels still contribute to the sum and denominator

and stay in numerical exports. Hiding never changes navigation history; in Fractions mode, hiding a channel also keeps the axes. Excluding retains the Show preference but disables that box until the channel is included again. For each measurement, a fraction is that selected channel's corrected yield divided by the sum of all selected corrected channels. Channel labels combine the integer index and optional text. Colours identify channels; source line/marker styles identify folders. Folder controls select among sources included in the main viewer. An unchecked main folder cannot contribute. Equal scan coordinates remain separate points with line breaks; neither energy overlap nor folder identity causes pooling.

If main false subtraction is enabled, every channel subtracts the corrected reference yield multiplied by its sampled-bin count divided by the reference bin count, matching the main viewer. A false-coincidence window without measured bins produces missing corrected values and a warning. Overlapping selected windows count shared bins in both channels; the window warns explicitly rather than changing the definitions.

The sum follows acquisition-time normalization and folder multipliers. Fractions are computed before the common folder multiplier, so even a zero display scale does not destroy a defined fraction. Negative corrected yields are retained. Missing channels, zero totals and numerically unresolved cancellation totals produce NaN fractions. **Fix fractions: -0.1 to 1.1** is on for a new analysis and holds that display range during zoom/pan/history and later updates; values outside it are not drawn and no warning counts them. Out-of-range values remain unchanged in calculations and exports; the checkbox limits their visibility only. Turn it off to fit all included channels, including hidden ones, from the selected folders; near-zero corrected totals can make that range very large. The status identifies these situations. Both intensity axes are always linear; the main Linear/Log setting does not apply here.

In Fractions mode the sum axis is labelled **Sum TPES** for threshold photon-energy data, **Sum photoionization spectrum** for all-electron energy data and **Summed ion yield** otherwise; hiding the sum hides that axis. The sum is over selected channels, not over the entire TOF range or over different scans.

The plot names folders in its upper-left legend (line style and marker). Channel colours match the coloured names in the list; **Channel legend** adds a legend of the shown channels, so that an image saved with the toolbar names them. Moving the mouse over a trace inside the plot draws it thicker, shows its channel or Sum, folder, scan coordinate and value next to the nearest data point, and highlights its row in the list. Where a channel and the sum overlap, the channel is shown. Hover is off while pan or zoom is active.

The toolbar provides rectangle zoom, pan, Back, Forward and Home. The mouse wheel zooms about the pointer in the shared scan axis and the shown intensity axes; a fixed fractional range stays locked. Shift limits wheel zoom to the scan axis; Ctrl limits it to the intensity axes. Toolbar rectangle zoom and pan act on both visible intensity scales, with the fractional lock respected. Home shows the whole scan of the selected folders. In Fractions mode it fits the fractional axis to all included channels, hidden ones too, or restores -0.1 to 1.1 when fixed; in Absolute mode it fits the shown traces. Home, channel Include changes, folder changes and fraction locking retain Back/Forward history. A changed scan coordinate or counts/count-rate unit, or a Fractions/Absolute switch, resets incompatible history. If no selected folder has a finite scan coordinate, the previous scan range stays visible with a status explanation. **Export > My traces...** writes one row per measurement of the folders selected in My traces, whatever the plot mode: the scan coordinate, `sum_yield`, one `yield[...]` and one `fraction[...]` column per included channel, then the folder, source ID, original index row, file, timestamp and acquisition time. Channel windows and corrections are in its metadata (see *Exporting results*). All relevant settings are saved in the workspace.

# Troubleshooting

- Restart the viewer after replacing its Python file; an already running window continues to use the old code.
- A `Non-numeric index row` message usually means that an older viewer is trying to read the new `MeasStart` timestamp or that the header and data columns do not match.
- `MeasStart contains no recognized date/time values` for a time-sliced export means that a PIMP release older than 18 October 2026 is reading it; open the folder with this release.
- An empty parameter is valid and is represented as unavailable data. If every value in a column is empty, verify that its channel exists in the measurement file.

- If no spectra appear, confirm that the index and corresponding TOF files are in the same folder and share the same measurement base names.
- If clicking or dragging on the matrix does nothing, the toolbar zoom or pan tool is still active. Click the highlighted tool again.
- If **Pick peaks** leaves the list empty, read the status line at the bottom of the Peaks window. Lower **Noise × σ** or **Min prominence / %**, untick **Visible spectrum only**, or choose **Clear selection** in the main window to search the sum of all measurements instead of one row. **Subtract false coincidences** becomes available after you drag or type a False coincidence window.
- Peak indices and m/z values come from the one-point **Calib** pair (TOF in µs and its m/z). Enter a pair for a known ion before picking peaks.