py_diAID User Guide

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

This step-by-step guide presents instructions for the installation and usage of py_diAID.

Table of Contents

Description

Data-independent acquisition coupled with parallel accumulation – serial fragmentation (dia-PASEF) has been gaining increasing traction, amongst proteomics researchers over the last years. dia-PASEF offers comprehensive proteome coverage, a high degree of reproducibility, and quantitative accuracy while using a much higher ion beam proportion than conventional DIA methods. Previous tools generated dia-PASEF methods with equidistant isolation widths and necessitated manual adjustment of the window design to the precursor density cloud. We present py_diAID, a Python-based package for Data-Independent Acquisition offering an Automated Isolation Design. py_diAID generates optimal dia-PASEF methods with variable isolation widths adjusted to the precursor density in *m/z* and automatically, optimally placed in the *m/z* – ion mobility (IM) plane. Variable isolation widths enable short acquisition cycles while covering essentially the complete *m/z*-IM-range. We found that the dia-PASEF methods, generated with py_diAID, are beneficial for optimizing proteomics workflows based on cell lines

(HeLa) or clinical samples such as CSF and Plasma, as well as for studying post-translational modifications such as phosphorylation.

We offer py_diAID as a Python module, command-line interface (CLI) tool, and Graphical User Interface (GUI) on all major operating systems under an Apache 2.0 license. py_diAID generates dia-PASEF methods with an optimal window design. It also allows for quality control of the precursors' distribution of a dataset in the *m/z*-IM plane and evaluating the suitability of already existing dia-PASEF methods for the individual experiment.

Installation

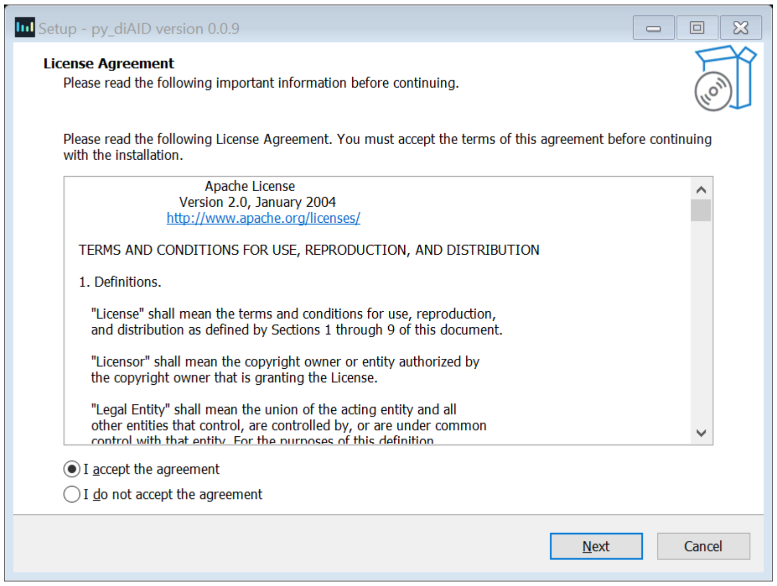
We offer py_diAID as a browser-based GUI, a stand-alone CLI and a Python package. All modes are also described on the GitHub repository (https://github.com/MannLabs/pydiaid).

Windows

Windows 10 is required to install py_diAID, and you need admin privileges if you want to run py_diAID for all users (right-click on the py_diAID logo on your desktop and select "Run as administrator"). We recommend uninstalling any previous py_diAID versions before updating py_diAID to prevent installation errors.

1. Download the latest release of the package for Windows (pydiaid_gui_installer_windows.exe) from the GitHub repository and execute the .exe file.
2. If the "User Account Control" dialog asks you for permission for the app to make changes to your device, please press the "Yes" button.
3. We recommend selecting "Install for me only (recommended)" if a "Select Setup Install Mode" dialog window shows up.

4. Please accept the License Agreement and click “Next” in the “Setup – py_diAID version X.X.X” dialog window.



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

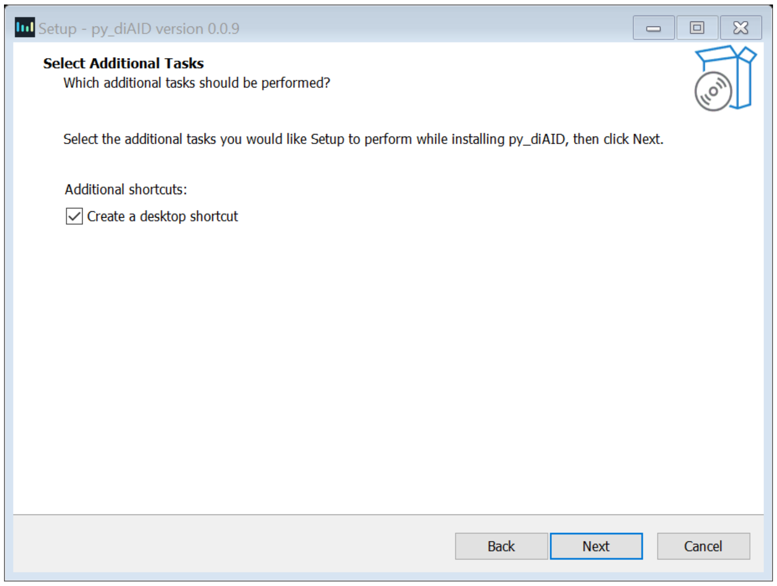
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition

[✓ I accept the agreement] I do not accept the agreement

5. In the “Select Additional Tasks” dialog window, activate the “Create a desktop shortcut” check box and click the “Next” button.



Select Additional Tasks

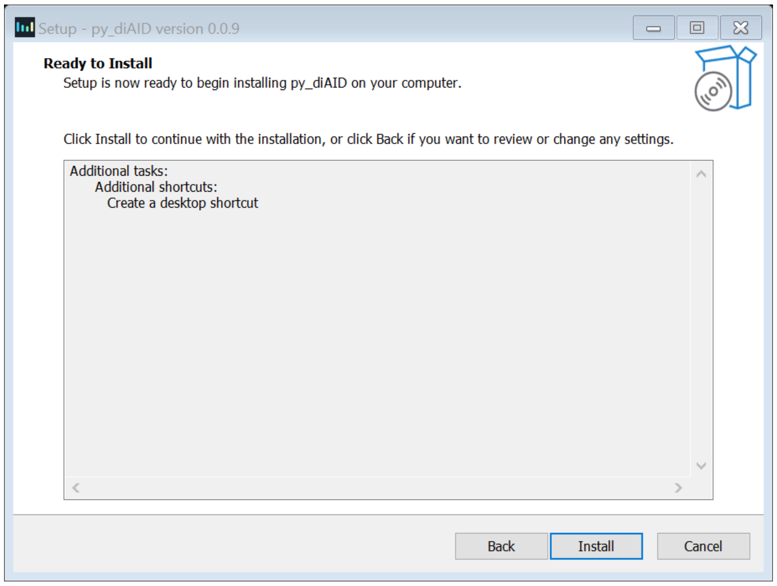
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing py_diAID, then click Next.

Additional shortcuts:

[✓] Create a desktop shortcut

6. Check the settings and press “Install”. You can change the setting with the “Back” or “Cancel” button prior to the installation.


Setup - py_diAID version 0.0.9

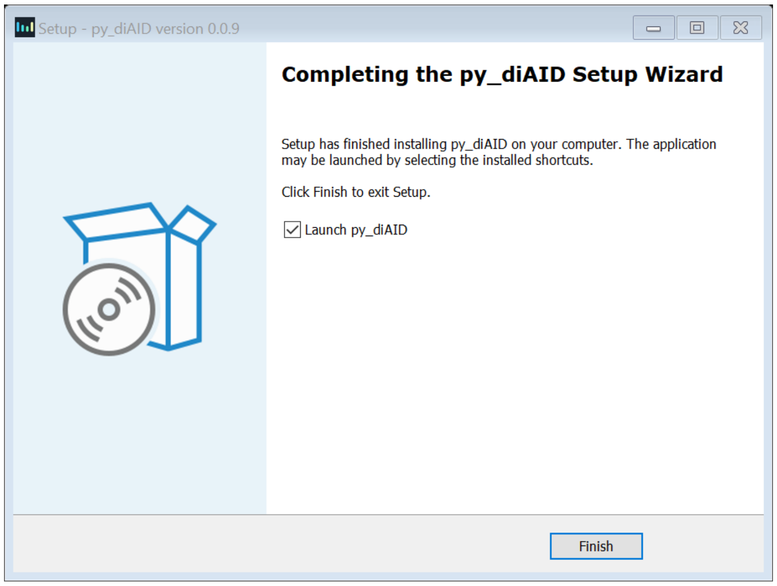
Ready to Install

Setup is now ready to begin installing py_diAID on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Additional tasks:
    Additional shortcuts:
      Create a desktop shortcut
```

7. Please wait until the installation is finished, activate the “Launch py_diAID” box, and click “Finish”.


Setup - py_diAID version 0.0.9

Completing the py_diAID Setup Wizard

Setup has finished installing py_diAID on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch py_diAID

8. If a “Window Security Alert” window appears, press “Allow access” to prevent the Windows Defender Firewall from blocking py_diAID on your PC.

MacOS

macOS Big Sur (11) pr higher is required to install py_diAID.

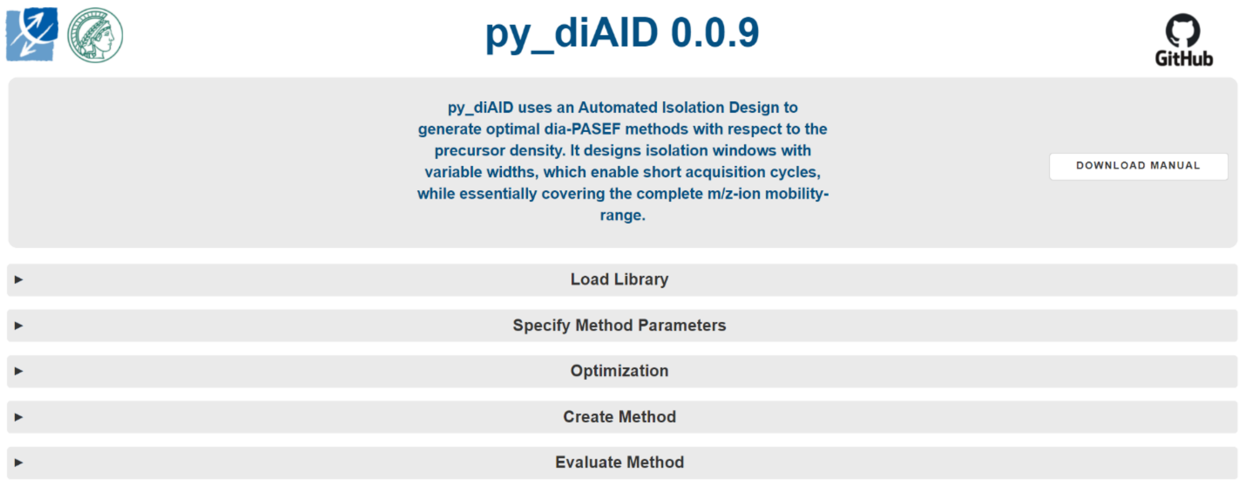
1. Download the latest release of the package for macOS (pydiaid_gui_installer_macos.pkg) from the GitHub repository and open the .pkg file. In rare cases, an error message shows up, which prevents the file from opening due to insufficient priveleges In this case, you can close the message by clicking “OK”, go to the “Security & Privacy” section of the “System Preferences” menu and press “Open Anyway” at the “General” tab.
2. Click “Continue” on the “Install py_diAID X.X.X” dialog window.
3. The License sections will show the Software License Agreement (Apache License 2.0). To continue the installation, press "Continue" button, and in the pop-up window you need to agree with the regulations of the license.
4. Please press “Install” to start the installation. This might take several minutes.
5. After the installation is finished, click “Close”. py_diAID is now available in the applications folder on your MacOS.

Linux

y_diAID can be used with Linus by installing it as a Debian package, which requires the Apache License 2.0’s acceptance.

1. Download the latest release of the package for Linux (pydiaid_gui_installer_linux.deb) from the GitHub repository.
2. Run the installer either by double clicking it, or by executing the command <*sudo dpkg -i pydiaid_gui_installer_linux.deb*> (copy everything between <>).

How to use py_diAID



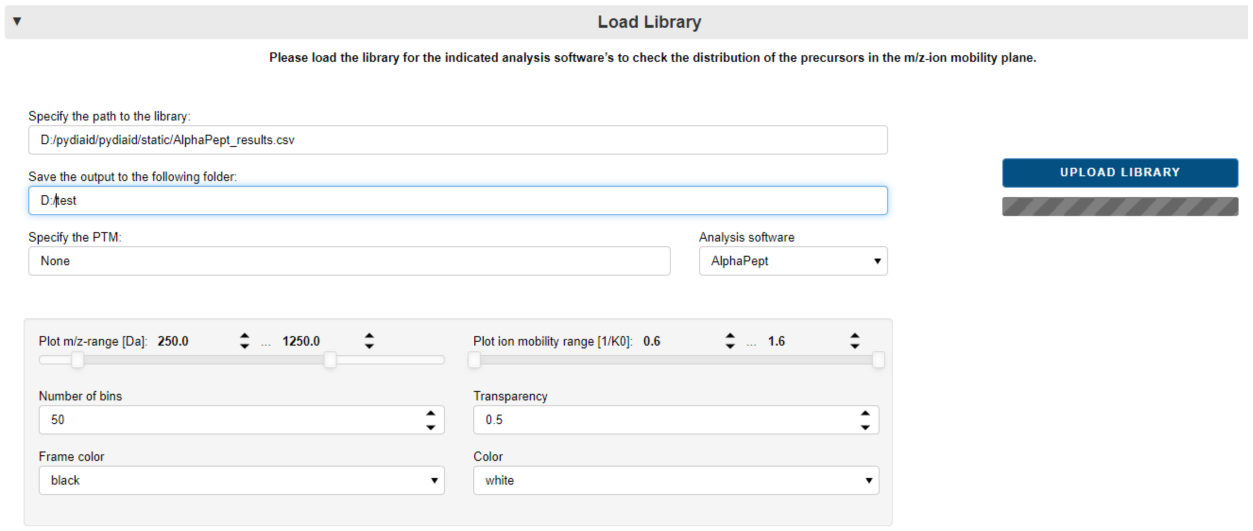
py_diAID 0.0.9

py_diAID uses an Automated Isolation Design to generate optimal dia-PASEF methods with respect to the precursor density. It designs isolation windows with variable widths, which enable short acquisition cycles, while essentially covering the complete m/z-ion mobility-range.

Launching py_diAID opens a terminal window displaying all the background information of py_diAID and a new browser tab called "py_diAID X.X.X" in your default browser. py_diAID can be terminated by either closing the tab or pressing "Ctrl+c" in the running terminal window.

We advise using either Google Chrome or Mozilla Firefox for optimal performance. However, the tool itself does not require an internet connection since py_diAID runs on your local system.

Load Library



Load Library

Please load the library for the indicated analysis software's to check the distribution of the precursors in the m/z-ion mobility plane.

Specify the path to the library: D:/pydiaid/pydiaid/static/AlphaPept_results.csv

Save the output to the following folder: D:/test

Specify the PTM: None

Analysis software: AlphaPept

Plot m/z-range [Da]: 250.0 ... 1250.0

Plot ion mobility range [1/K0]: 0.6 ... 1.6

Number of bins: 50

Transparency: 0.5

Frame color: black

Color: white

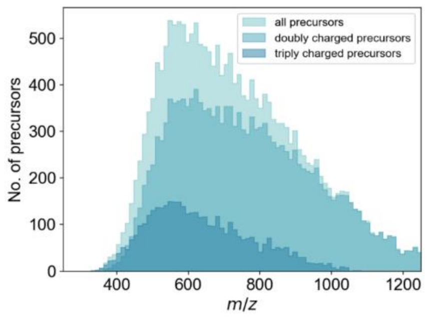
- Provide the file path to the library (*i.e.* proteomics dataset) analyzed by AlphaPept, MaxQuant, Spectronaut or FragPipe. **Note:** Specify the software in the drop-down menu "Analysis software".
- Specify the distinctive characters of the post-translational modification that interests you. This function filters the modified peptide name column for the specified

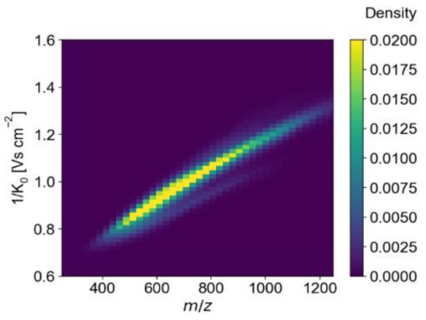
characters. For Spectronaut, “STY” would be suitable for filtering phosphopeptide precursors and FragPipe “UniMod:28”.

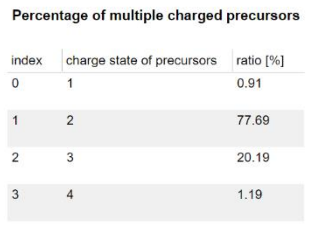
- The following setting will be applied to all plots that are generated by py_diAID: “Plot I-range” and “Plot ion mobility-range” indicate the limits of the *m/z* and ion mobility axis in all plots. py_diAID calculates and plots kernel density estimations to display the precursor distribution. The “Number of bins” has an influence on the resolution of these plots. We recommend 50 bins for time efficiency and, for sharper figures, a resolution of up to 300 bins. However, this will take up to 30 minutes. “Transparency”, “Frame color”, and “Color” are properties of the dia-PASEF windows that are plotted to visualize the dia-PASEF methods.

A test library can be found on the following path: <*py_diAID installation directory*>\pydiaid\pydiaid\static\AlphaPept_results.csv. It represents a 200ng tryptic HeLa digest acquired with 21-minute Evosep gradients and dda-PASEF.

Press the “Upload Library” button to visualize the precursor distribution of the library/dataset with a histogram over *m/z*, a kernel density plot, and the charge state ratios. This is the only mandatory step to execute the “create” and “evaluate” buttons.






Percentage of multiple charged precursors

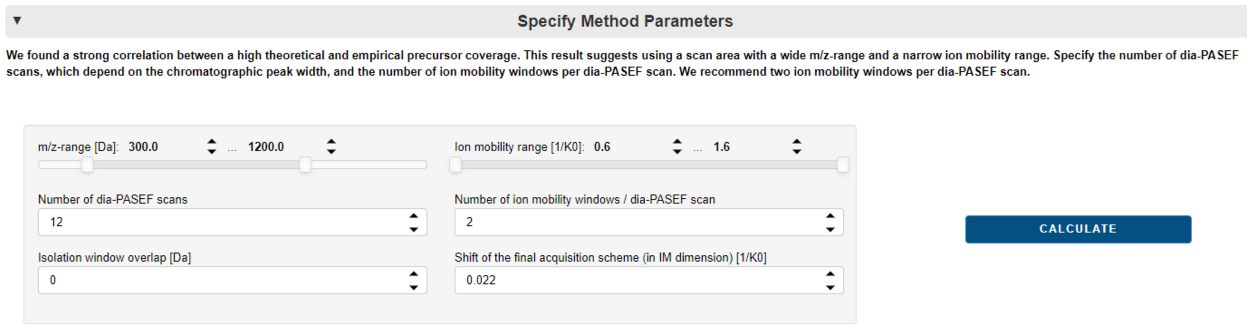
index	charge state of precursors	ratio [%]
0	1	0.91
1	2	77.69
2	3	20.19
3	4	1.19

Analysis software instructions

- *AlphaPept:* Please use the **results.csv** file. py_diAID uses the columns “q_value”, “decoy”, “mz”, “mobility”, “charge”, “protein”, and “precursor”.
- *MaxQuant:* Please use the **evidence.txt** file. py_diAID uses the columns “Reverse”, “Potential contamination”, “m/z”, “1/K0”, “Charge”, “Proteins”, and “Modified sequence”.
- *Spectronaut single-run:* Please export the dataset in the **normal long** format already pre-filtered with “Quantification Data Filtering” and “No Decoy” as .tsv or .csv file. py_diAID requires the columns “'FG.PrecMzCalibrated”, “FG.ApexIonMobility”, “FG.Charge”, “PG.ProteinGroups”, and “EG.PrecursorId”.

- *Spectronaut library:* Please export the library as an**.xls file**. py_diAID requires the columns "PrecursorMz", "IonMobility", "PrecursorCharge", "UniProtIds", and "ModifiedPeptide".
- *FragPipe:* The dataset should be analyzed while spectral library generation is active. Upload the **library.tsv** file. py_diAID will use the columns "PrecursorMz", "PrecursorIonMobility", "PrecursorCharge", "ProteinId", and "ModifiedPeptideSequence".

Specify Method Parameters


Specify Method Parameters

We found a strong correlation between a high theoretical and empirical precursor coverage. This result suggests using a scan area with a wide m/z-range and a narrow ion mobility range. Specify the number of dia-PASEF scans, which depend on the chromatographic peak width, and the number of ion mobility windows per dia-PASEF scan. We recommend two ion mobility windows per dia-PASEF scan.

m/z-range [Da]: 300.0 ... 1200.0

Ion mobility range [1/K0]: 0.6 ... 1.6

Number of dia-PASEF scans: 12

Number of ion mobility windows / dia-PASEF scan: 2

Isolation window overlap [Da]: 0

Shift of the final acquisition scheme (in IM dimension) [1/K0]: 0.022

The following parameters define the optimal dia-PASEF method:

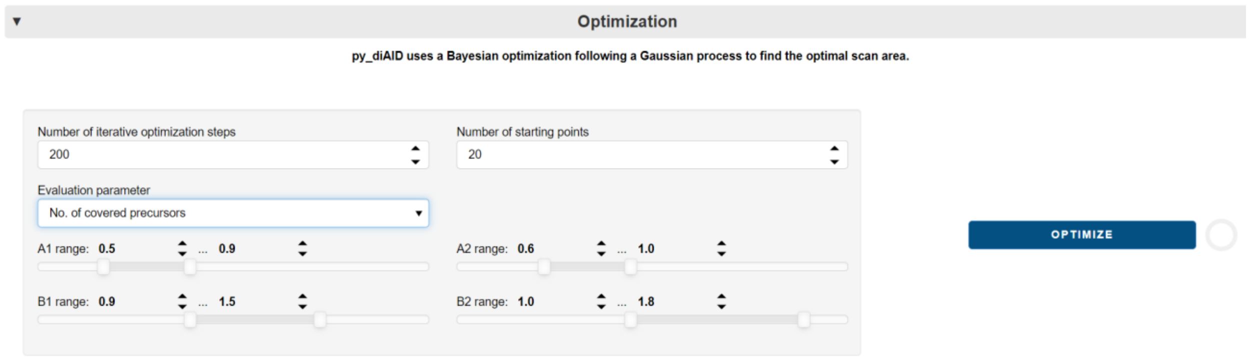
- ***m/z*-range:** Variable isolation windows enable the coverage of a wide *m/z*-range while keeping the cycle time short. Additionally, we found a strong correlation between theoretical and empirical precursor coverage. Therefore, we recommend using wide *m/z*-ranges for generating dia-PASEF methods.
- **Ion mobility range:** we recommend a reduced IM range.
- **Number of dia-PASEF scans:** The number of dia-PASEF scans influences the cycle time. Since the ramp and accumulation time is set to 100ms by default, you can calculate the cycle time as follows: 100 ms (for each MS1 scan) + 100 ms x number of dia-PASEF scans + additional transfer time. We recommend 8 dia-PASEF scans for 11 min Evosep gradients, 12 dia-PASEF scans for 21 min Evosep gradients, and 25 dia-PASEF scans for 44 min Evosep gradients (see "Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF" Experimental procedures).
- **Number of ion mobility windows / dia-PASEF scans:** One dia-PASEF scan can be separated into multiple ion mobility windows. The number of ion mobility windows defines how often the quadrupole changes its position. For two ion mobility windows per dia-PASEF scan, the quadrupole changes once its isolation window during one dia-PASEF scan. We recommend two ion mobility windows per dia-PASEF scan since this results in the highest precursor coverage based on simulations and in higher peptide identifications (see "Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF" Suppl. Figure S5).
- **Isolation window overlap:** This value defines the overlap in Da. We recommend no isolation window overlap for short cycle times.

- **Shift of the final acquisition scheme (in IM dimension):** The dia-PASEF acquisition scheme will be optimized for a maximized theoretical precursor coverage. However, in practice, the quadrupole has a blind spot while changing to the isolation width of the next ion mobility window during one dia-PASEF scan. This blind spot results in missed precursors in the top ion mobility window region. The blind spot has an ion mobility height of approximately 0.022 $1/K_0$. Therefore, we recommend to shift the final acquisition scheme upwards by 0.022 $1/K_0$ to not miss signal in the densest doubly charged precursor cloud.

Press “calculate” to activate all settings and to calculate the precursors, which fall within the selected scan area defined by the *m/z*- and IM ranges.

index	precursors within the scan area [%]
0	97.59

Optimization


▾ Optimization

py_diAID uses a Bayesian optimization following a Gaussian process to find the optimal scan area.

Number of iterative optimization steps: 200

Number of starting points: 20

Evaluation parameter: No. of covered precursors

A1 range: 0.5 ... 0.9

A2 range: 0.6 ... 1.0

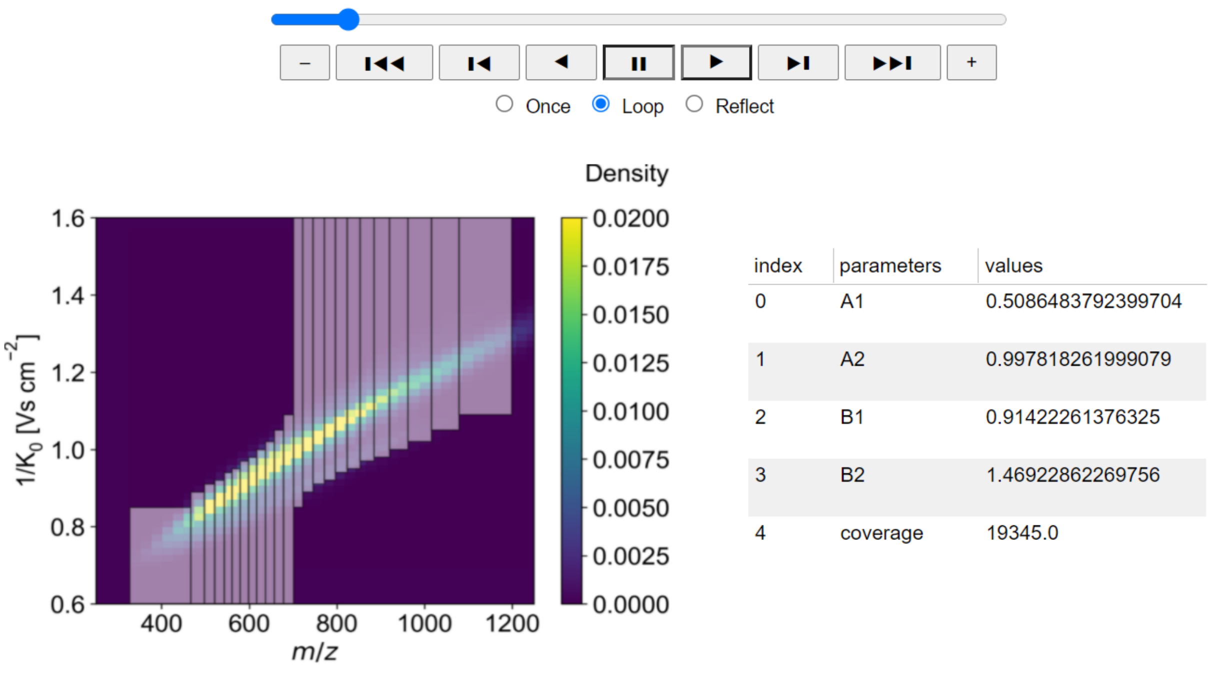
B1 range: 0.9 ... 1.5

B2 range: 1.0 ... 1.8

py_diAID calculates the isolation window width based on the precursor distribution in *m/z.* The position of the isolation windows in the *m/z* and IM plane is dependent on a trapezoid that reflects the scan area. py_diAID selects a set of random A1, A2, B1, and B2 coordinates that define the corners of the trapezoid. A1, A2, B1, and B2 coordinates must lie within the respective ranges. Next, it generates a dia-PASEF method based on the placement of the trapezoid and the specified method parameters. This step is followed by the evaluation of the developed dia-PASEF method. Depending on the evaluation result, py_diAID decides which trapezoid coordinates should be attempted next, to find the optimal precursor coverage. The decision is based on a Bayesian optimization following a Gaussian process offered by the skopt package. A detailed description of the py_diAID algorithm can be found in our publication: “Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF” Figure 2 and Suppl. Figure S6.

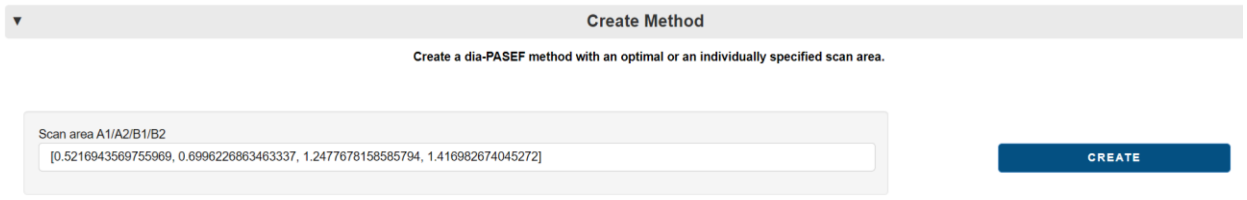
- **Number of iterative optimization steps:** This value defines the number of iterations to find the optimal position of the dia-PASEF acquisition scheme.
- **Number of starting points**: The algorithm initially tries out multiple random sets of coordinates until it considers the evaluation results to decide which parameters are reasonable to try next. **“**Number of starting points” defines the number of this initial random set. We recommend 200 iterative optimization steps and 20 starting points for a result that provides 99.2% reproducibility (see “Rapid and in-depth coverage of the (phospho-)proteome with deep libraries and optimal window design for dia-PASEF” Suppl. Figure S7).
- **Evaluation parameter:** py_diAID can use different factors to evaluate the coverage of the dia-PASEF methods. “Number of covered precursors” includes all precursors except singly charged precursors and is recommended.
- **A1/A2/B1/B2 range:** The coordinates A1, A2, B1, B2 of the trapezoid are selected from within these ranges. (Please find a more detailed description in Figure 2 and supplementary Figure S6 in our publication.)

Press the “Optimize” button to start the optimization process. py_diAID needs approximately 15 minutes for 200 iterative optimization steps. You can supervise all optimization steps after the optimization process is finished using the built-in player.



index	parameters	values
0	A1	0.5086483792399704
1	A2	0.997818261999079
2	B1	0.91422261376325
3	B2	1.46922862269756
4	coverage	19345.0

Create Method



▼ Create Method

Create a dia-PASEF method with an optimal or an individually specified scan area.

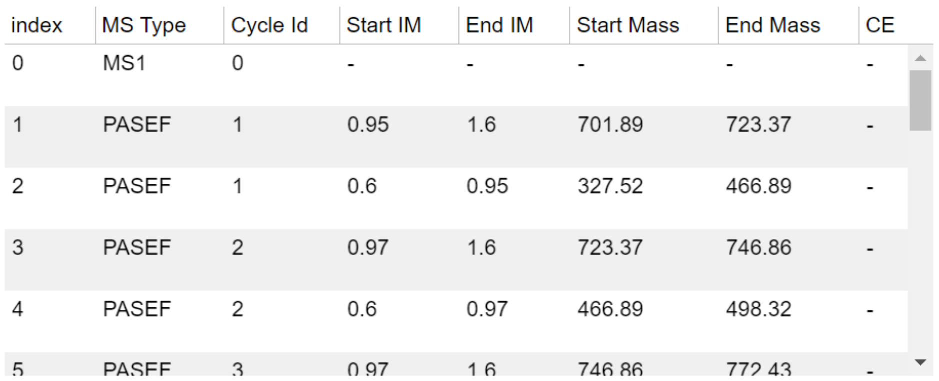
Scan area A1/A2/B1/B2

[0.5216943569755969, 0.6996226863463337, 1.2477678158585794, 1.416982674045272]

CREATE

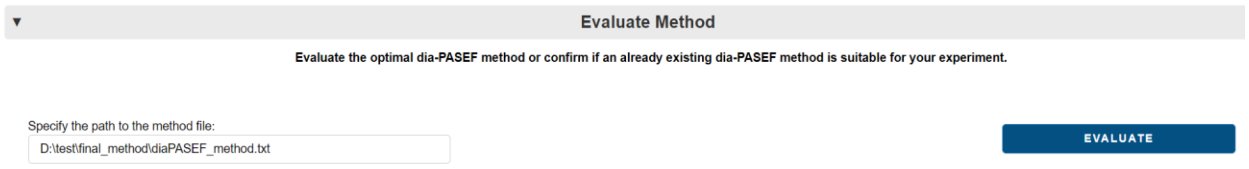
py_diAID writes the optimal trapezoid coordinates in the file “Scan area A1/A2/B1/B2”. You can also skip the optimization process and replace these coordinates with different coordinates if you already have coordinates from a previous optimization process.

Press the “Create” button to generate the final dia-PASEF method based on the specified trapezoid coordinates. py_diAID shifts this method upwards in the IM by the value specified in “Shift of the final acquisition scheme”.



index	MS Type	Cycle Id	Start IM	End IM	Start Mass	End Mass	CE
0	MS1	0	-	-	-	-	-
1	PASEF	1	0.95	1.6	701.89	723.37	-
2	PASEF	1	0.6	0.95	327.52	466.89	-
3	PASEF	2	0.97	1.6	723.37	746.86	-
4	PASEF	2	0.6	0.97	466.89	498.32	-
5	PASEF	3	0.97	1.6	746.86	772.43	-

Evaluate Method



▼ Evaluate Method

Evaluate the optimal dia-PASEF method or confirm if an already existing dia-PASEF method is suitable for your experiment.

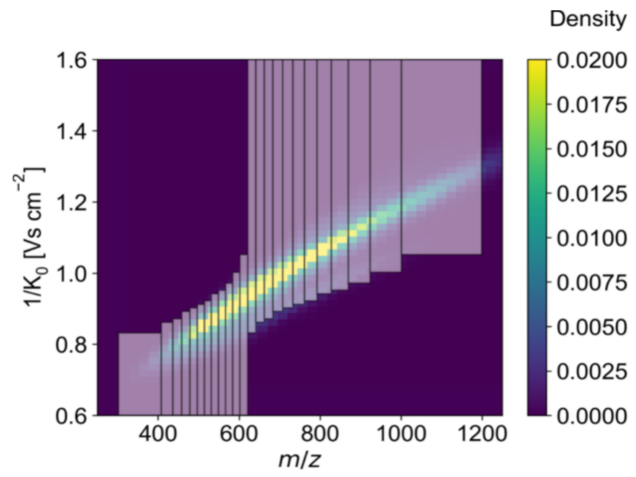
Specify the path to the method file:

D:\test\final_method\diaPASEF_method.txt

EVALUATE

py_diAID writes the file path of the final acquisition scheme automatically in the field “Specify the path to the method file”. You can also define an individual path to evaluate if an already existing dia-PASEF method is suitable for your experiment, based on the previously loaded library.

Press the "Evaluate" button to display the result of the evaluation



index	evaluation parameter	value
0	precursors within m/z-range [%]	97.59
1	unique proteins in the library	3420
2	unique precursors in the library	19998
3	smallest diaPASEF window	17.03
4	biggest diaPASEF window	198.07
5	average diaPASEF window size	37.4
6	No. of covered proteins	3397
7	No. of covered precursors	19257
8	No. of covered, doubly charged precursors	15056
9	No. of covered, triply charged precursors	3792
10	No. of covered, quadruply charged precursors	232
11	all proteins covered	99.3%
12	all precursors covered	96.3%
13	all doubly charged precursors covered	96.9%
14	all triply charged precursors covered	93.9%
15	all quadruply charged precursors covered	97.5%