Supplementary Information

Automatic recognition of Pd ions in high-resolution mass spectra of multicomponent samples without visible fine isotope structures

Pavel E. Gurevich,[1,2†] Konstantin S. Kozlov,[1†] Artem S. Silverstov,[1] Lev E. Nersesyan,[1] Nikita I. Kolomoets,[1] Valentina V. Ilyushenkova,[1] Julia V. Burykina,[1] Daniil A. Boiko,[1] Artem P. Vorozhtsov,[1] and Valentine P. Ananikov[1,*]

[1] *Zelinsky Institute of Organic Chemistry, Russian Academy of Sciences, Leninsky Prospekt 47, Moscow, 119991, Russia; https://ananikovlab.ru/*

[2] *Center for Energy Science and Technology, Skolkovo Institute of Science and Technology, Bolshoy Boulevard 30, bld. 1, Moscow, 121205, Russia*

† these authors contributed equally

* corresponding author, e-mail: val@ioc.ac.ru

Table of Contents

Supplementary Note 1 Deisotoping methods

Supplementary Note 1.1 Graph theory terms

Graph can be defined as the set of two subsets (*V, E*), where *V* contains so-called vertices or nodes and *E* contains edges, each of them can be considered as the transition between two nodes. In the most basic concept nodes can be represented by their individual numbers, but additional attributes can be added. Otherwise, edges may possess direction (directed graph) and weight (weighted graph). Vertex is called an incident, if it is one of the two vertices connected by the edge. During traversing through the vertices (directed or not) may be used such terms as child (vertex we move towards to) or ancestor (vertex, which we move from).

Path in the graph is an ordered sequence of vertices (v_1, v_2, … v_n), where for each adjacent pair of vertices there is an edge $\{v_i, v_{i+1}\}$ for i = 1, 2, …, n - 1. If the last vertex is connected with the first one in sequence, the path is called a cycle.

If every vertex in a graph can be reached from another vertex, the graph is called connected. In case of directed graphs, the division on weakly connected and strongly connected graphs takes place. If the graph becomes connected after replacement of all directed edges, the graph is called weakly connected. If every vertex can be reached despite the existence of directed edges, the graph is strongly connected.

Connected component in an undirected graph is a maximal connected subgraph. In the case of directed graphs, the division on strongly connected components and connected components takes place. Connected graph without cycles is called a tree. Starting point of traversing in a tree is called a root.

Depth-first search (DFS) algorithm is based on - how it could be noticed - on search in depth. First, we initialize a list of visited nodes. At the beginning, it is filled with zeros (False). Then, we start from some node (root), mark it as visited and iterate through every adjacent node. If this vertex was not visited, we repeat all actions from the previous sentence. Therefore, we go through all “branches” in the graph.

Supplementary Note 1.2 Graph-based deisotoping algorithm implementation details

In graph-based deisotoping implementation each node is described by a unique number, peak *m*/*z* and group number; edges contain information about two incident vertices numbers, charge and weight, which is equal to $|m_j - m_i - \frac{\Delta m}{z}| < \delta$. Therefore, we obtain a weighted directed graph, which consists of tree components (or a forest).

First, initialization of the list of used vertex indicators (visited) and list with each vertex group number (groups) is performed.
Due to the importance of detecting multiple charged ions, the loop starts from the highest charge value to the lowest. In each iteration, we start going from a vertex with the lowest m/z value and apply depth-first search (DFS). The recursive function finds all possible paths in a tree with a root in our current node. It saves information about the current vertex number and iterates through every incident node with the same charge. In case of an unvisited child, DFS starts from it and adds the weight of the edge to a special counter. If the algorithm didn't find any children of vertex, it saves information about vertices in the path and the mean value of edge weights (paths with only 2 vertices and multiple charges are not considered). After returning to the root, the path with the lowest mean value is chosen. For each vertex in the path the same number in the groups is given and all of them are labeled in visited. These iterations continue until all vertices are labeled.

In the graph-based deisotoping implementation nodes are not labeled as visited immediately due to the possibility of not being used for clustering in this exact iteration. After the application of DFS on one of the components (which can be defined as the trees), we obtain the list of branches with their average weight, a branch with the least value of weight is chosen and all its vertices are marked as used. Pseudocode is shown below.

Supplementary Box 1. Mass spectrum deisotoping algorithm

```
function RUN(spectrum, charged = False, **find_peak_kwargs)
    Input: spectrum - mass spectrum, charged - charge flag, find_peak_kwargs - peak finding parameters
    Output: array of predictions (regular or charged)
    peaks ← find_peaks(spectrum, find_peak_kwargs)
    m ← peaks[0]                                                    ▷ Peak masses
    n ← length(m)                                                   ▷ Number of peaks
    adj_list ← make_links(spectrum, find_peak_kwargs)               ▷ Adjacency list
    Initialize arrays:
    preds ← array of zeros with size spectrum.masses filled with -1
    charged_preds ← array of zeros with size spectrum.masses filled with -1
    deisotoped ← array of zeros with size n filled with -1
    deisotoped_charged ← array of zeros with size n filled with -1
    used ← list of zeros with length n                              ▷ Peak usage flags
    paths ← empty list                                              ▷ List of found paths
    counter ← 0                                                     ▷ Cluster counter
    for z from max_charge down to 1 do
        for i from 0 to n − 1 do
            if used[i] = 1 then
                continue
            end if
            SEARCH(adj_list[i], charge=z, path="", mean=0.0)
            if paths is empty then
                continue
            end if
            Sort paths by descending mean value
            chosen_path ← last element of paths
            Remove chosen_path from paths
            if chosen_path[0] is empty then
                continue
            end if
            for each v in chosen_path[0] do
                deisotoped[v] ← counter
                deisotoped_charged[v] ← chosen_path[−1]
                used[v] ← 1
            end for
            counter ← counter + 1
            Clear paths
        end for
    end for
    preds[peaks[2]] ← deisotoped
    Replace all −1 in deisotoped_charged with 1
    deisotoped_charged ← normalize_labels(deisotoped_charged)
    charged_preds[peaks[2]] ← deisotoped_charged
    if charged = False then
        return preds
    else
        return charged_preds
    end if
end function
```

Supplementary Box 1. Mass spectrum deisotoping algorithm (continued)

```
function SEARCH(node, charge = −1, path = "", mean = 0.0)
    curr_path ← path + str(node[0]) + ""
    c ← 0                                                    ▷ Child counter
    if node[1] is not empty then                  ▷ If there are adjacent nodes
        for each nd in node[1] do
            if used[nd[1]] = 0 and nd[0] = charge then
                c ← c + 1
                SEARCH(adj_list[nd[1]], charge=nd[0], path=curr_path,
mean=mean + nd[2])
            end if
        end for
    end if
    if c = 0 then                                  ▷ If end of path is reached
        if charge ≤ 1 then
            mean ← mean/length(curr_path.split())
            Add (curr_path.split(), mean, 1) to paths
        end if
        if charge > 1 and length(curr_path.split()) > 2 then
            mean ← mean/length(curr_path.split())
            Add (curr_path.split(), mean, charge) to paths
        end if
    end if
end function
```

Supplementary Note 1.3 Synthetic spectra generation for deisotoping model evaluation

25 synthetic spectra for comparison of models were generated with the possible number of isotopic distributions ranging from 100 to 200 with the probability of the ion being multiply-charged equal to 20%.

In order to implement a tree-ensemble-based deisotoping model applicable on simple HRMS spectra to compare with the developed graph-based approach, the generation of ≈100k dataset of pairs of peaks was performed. The training dataset is generated by synthesizing artificial mass spectra. First, noise is created based on pre-recorded spectra, then isotopic distributions are computed with added mass error and random intensity scaling to simulate realistic conditions and varying concentrations. Each peak is assigned a substance label. Finally, pairs of peaks are formed and given binary labels indicating whether they belong to the same isotopic distribution or not.

Supplementary Note 1.4 Deisotoping metrics explanation

Individual isotopic distribution extraction can be formulated as a detection problem. A good deisotoping model should detect the object (isotopic distribution) fully, but it is also necessary to avoid combining peaks from different distributions into the one detected object. An established method to evaluate object detection quality of the model is a calculation of AveragePrecision@k.

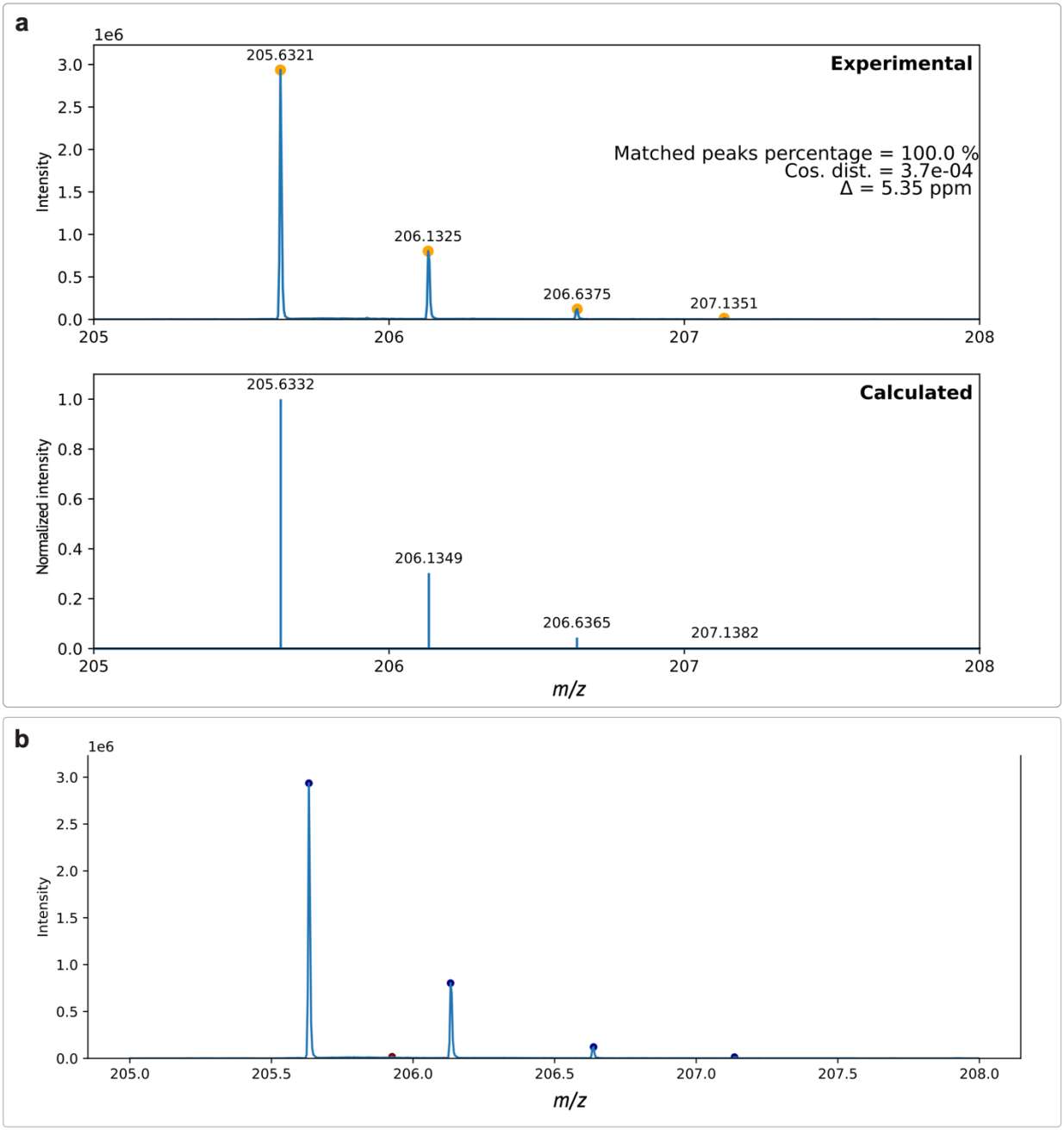
Applied to deisotoping, we calculate Precision@k for each multicomponent spectrum in the test set by the rule below:

1. For each true isotopic distribution in the spectrum, we pick the most intensive peak and choose predicted isotopic distribution, where this peak is contained. Then, calculation of Intersection over Union (IoU) score is performed between true and predicted isotopic distributions.
2. If IoU score is bigger than the determined parameter *k* (e.g. 0.25, 0.50, 0.75), then isotopic distribution is considered to be correctly detected.
3. Precision@k is calculated as the relative number of correctly detected isotopic distributions.
4. Spectrum-averaged Precision@k gives us AveragePrecision@k.

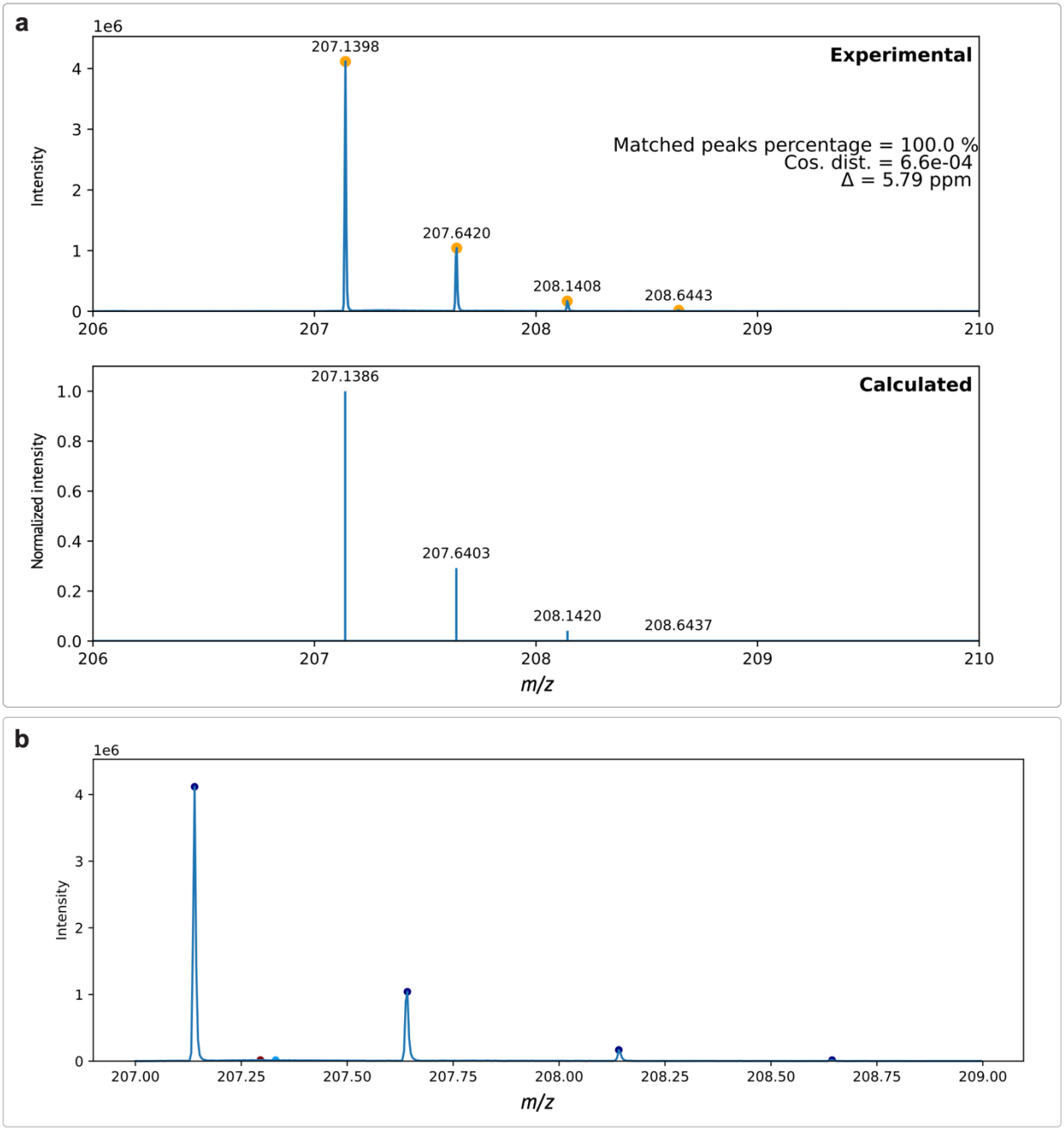
We used the best settings to record high quality mass spectra. However, in some cases artefacts may appear. On some spectra low intensity signals were also processed and marked. However, we do not use them for making conclusions in the study.

Supplementary Note 1.5 Graph-based deisotoping performance on real spectra

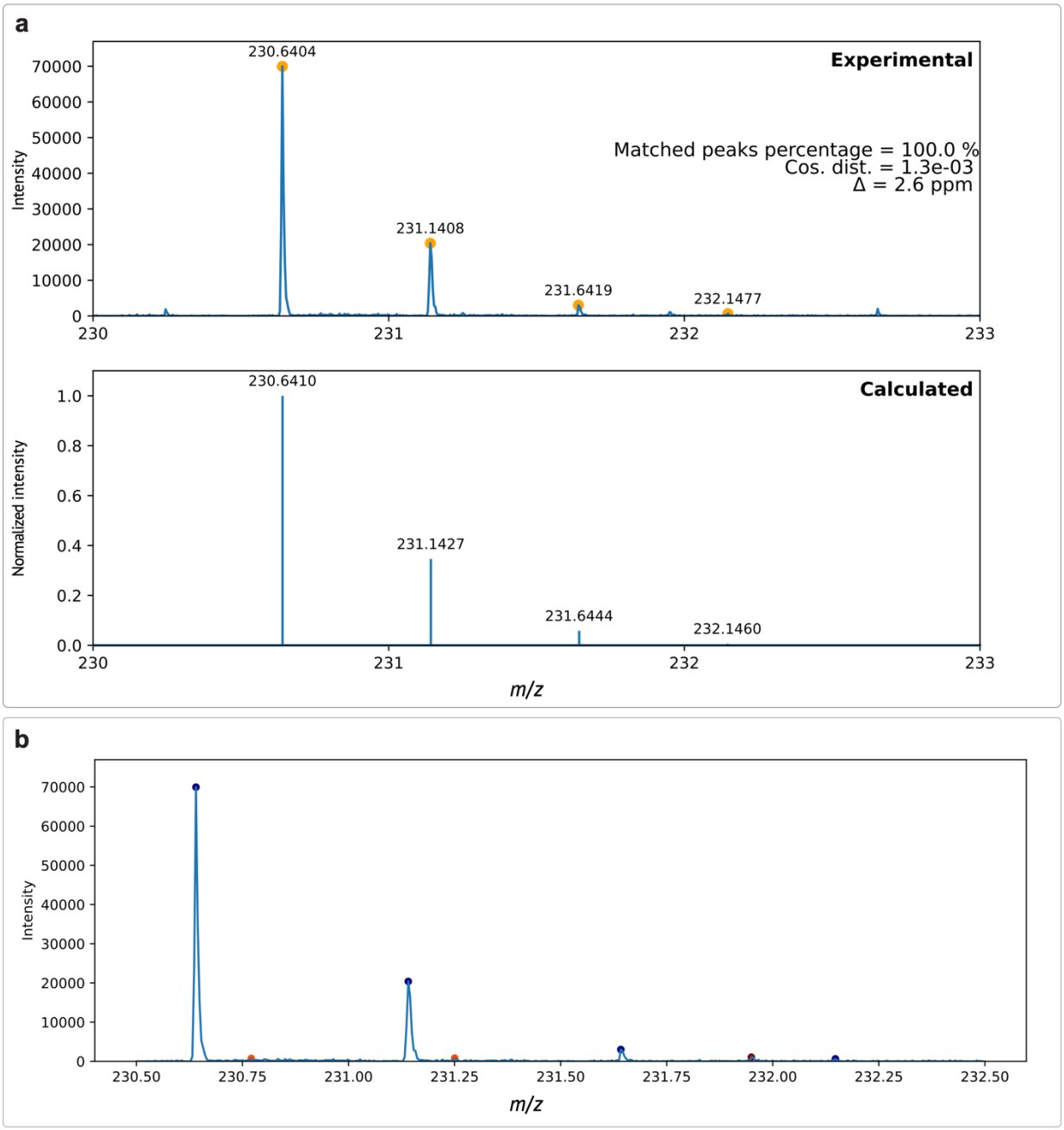
Here, graph-based deisotoping algorithm performance on real spectra is presented. Only 12 double-charged ions are shown. Results of 88 single charged ions deisotoping used can be found on Figshare.[1]



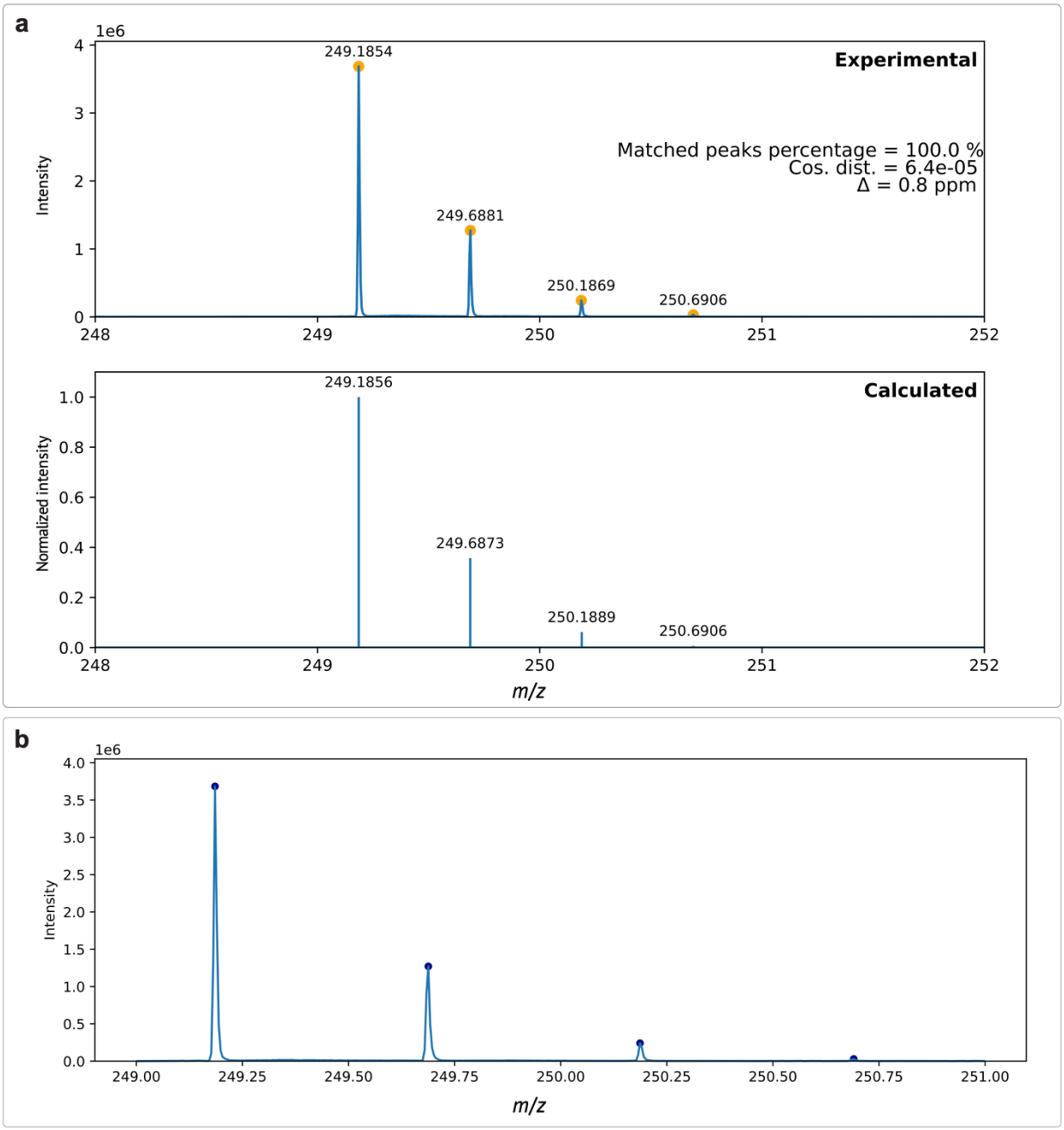
Supplementary Figure 1. Graph-based deisotoping performance on complex spectrum, which contains $C_{28}H_{33}N_3^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



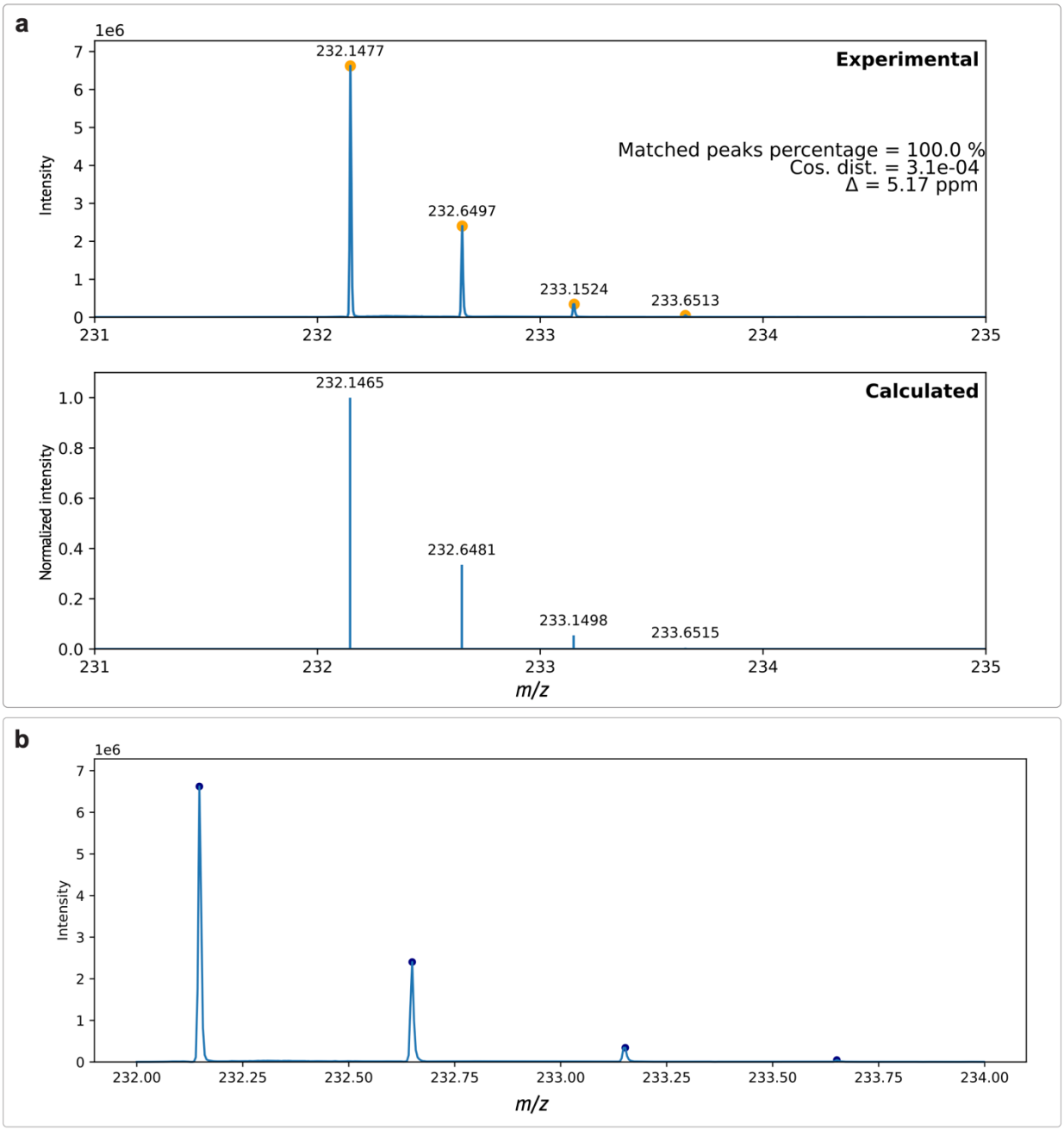
Supplementary Figure 2. Graph-based deisotoping performance on complex spectrum, which contains $C_{27}H_{34}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



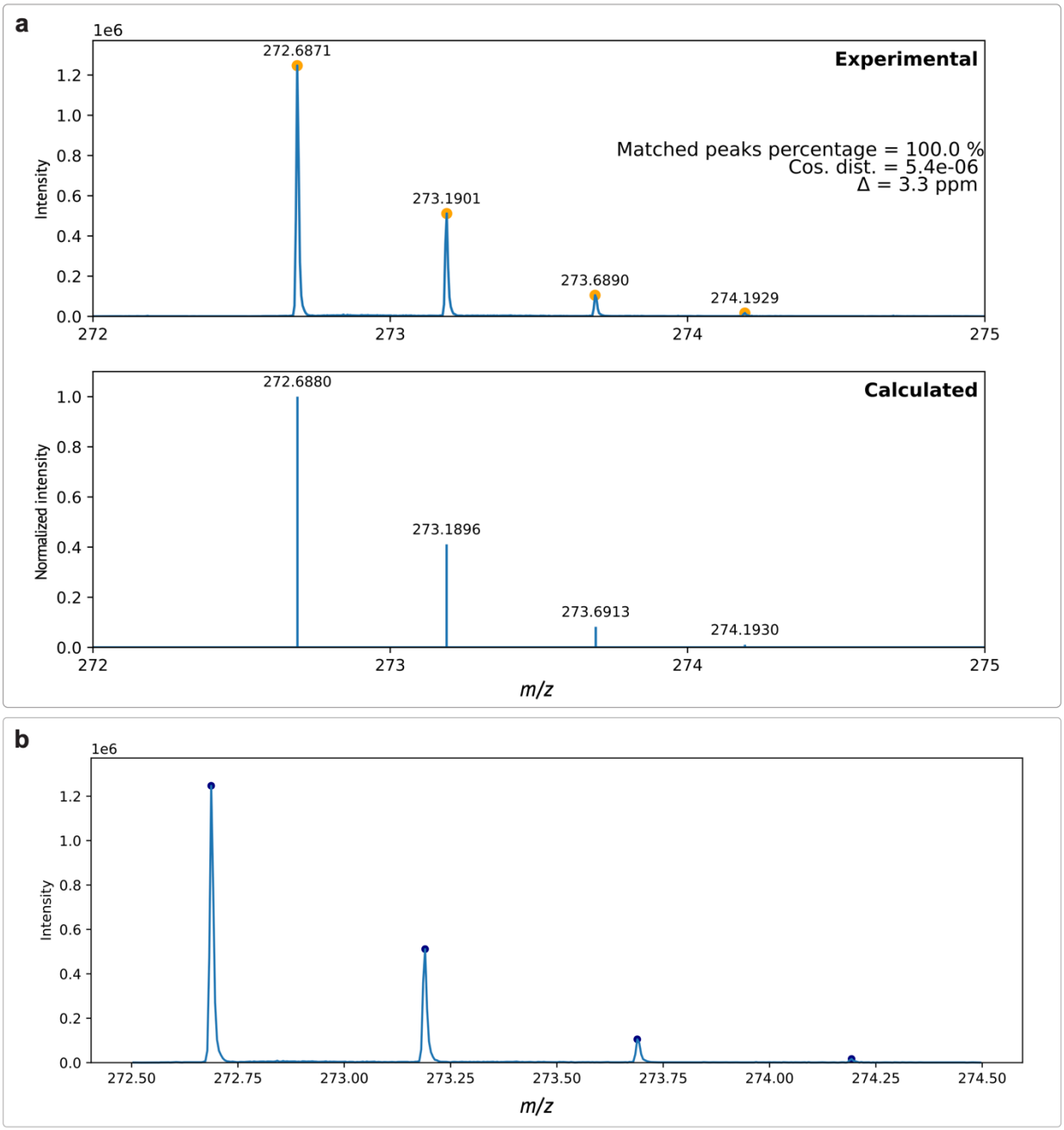
Supplementary Figure 3. Graph-based deisotoping performance on complex spectrum, which contains $C_{32}H_{35}N_3^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



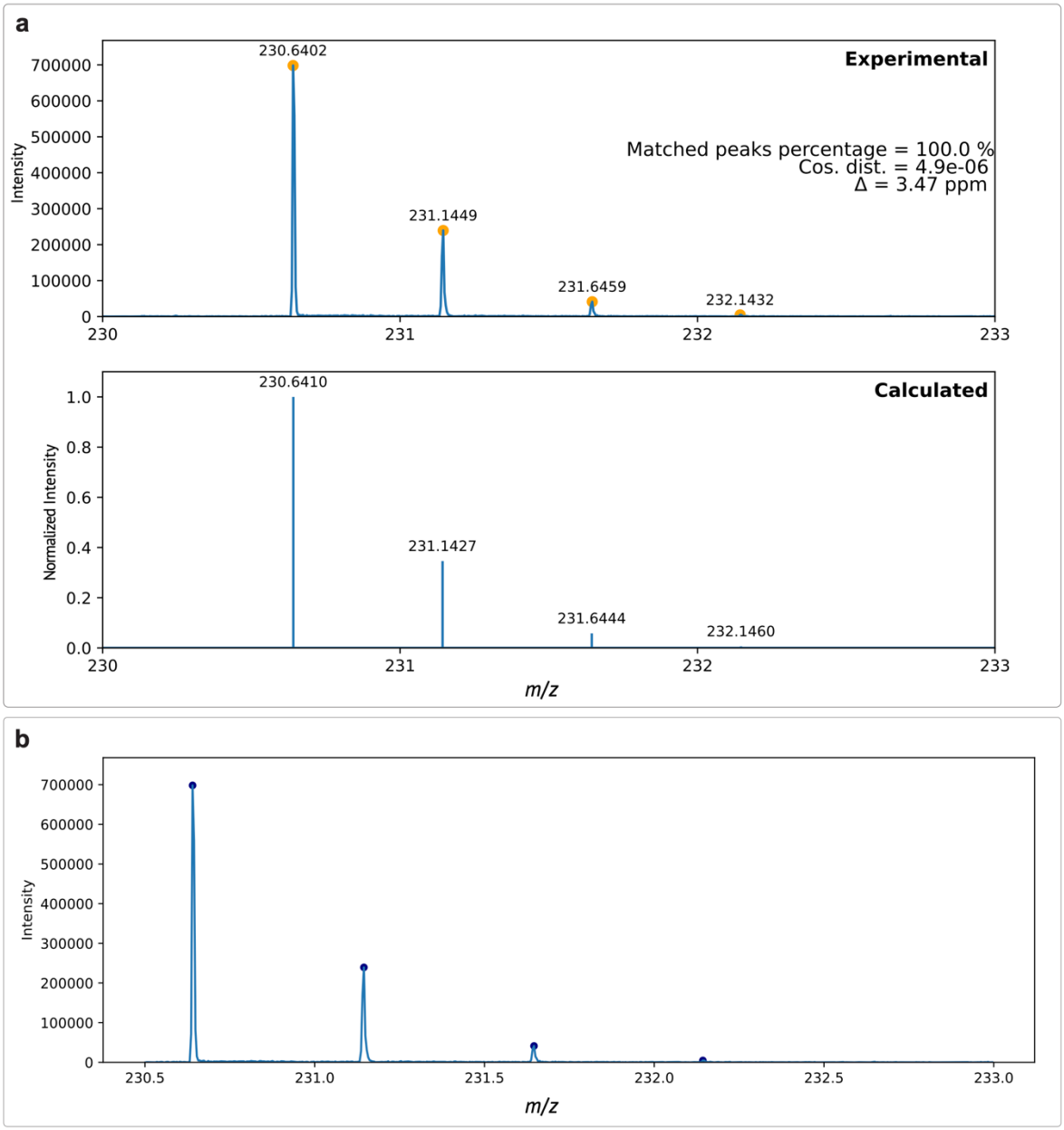
Supplementary Figure 4. Graph-based deisotoping performance on complex spectrum, which contains $C_{33}H_{46}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



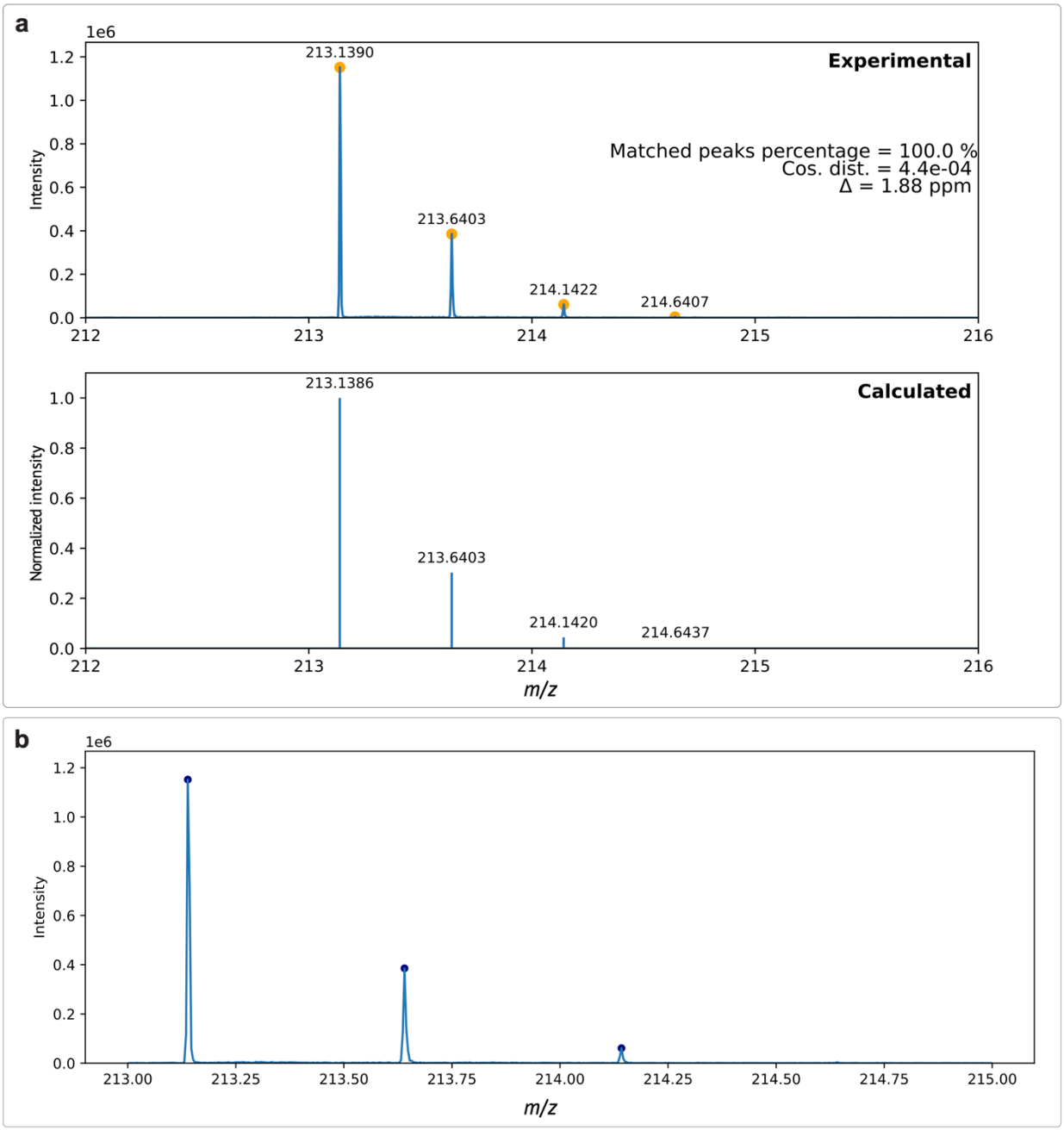
Supplementary Figure 5. Graph-based deisotoping performance on complex spectrum, which contains $C_{31}H_{36}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



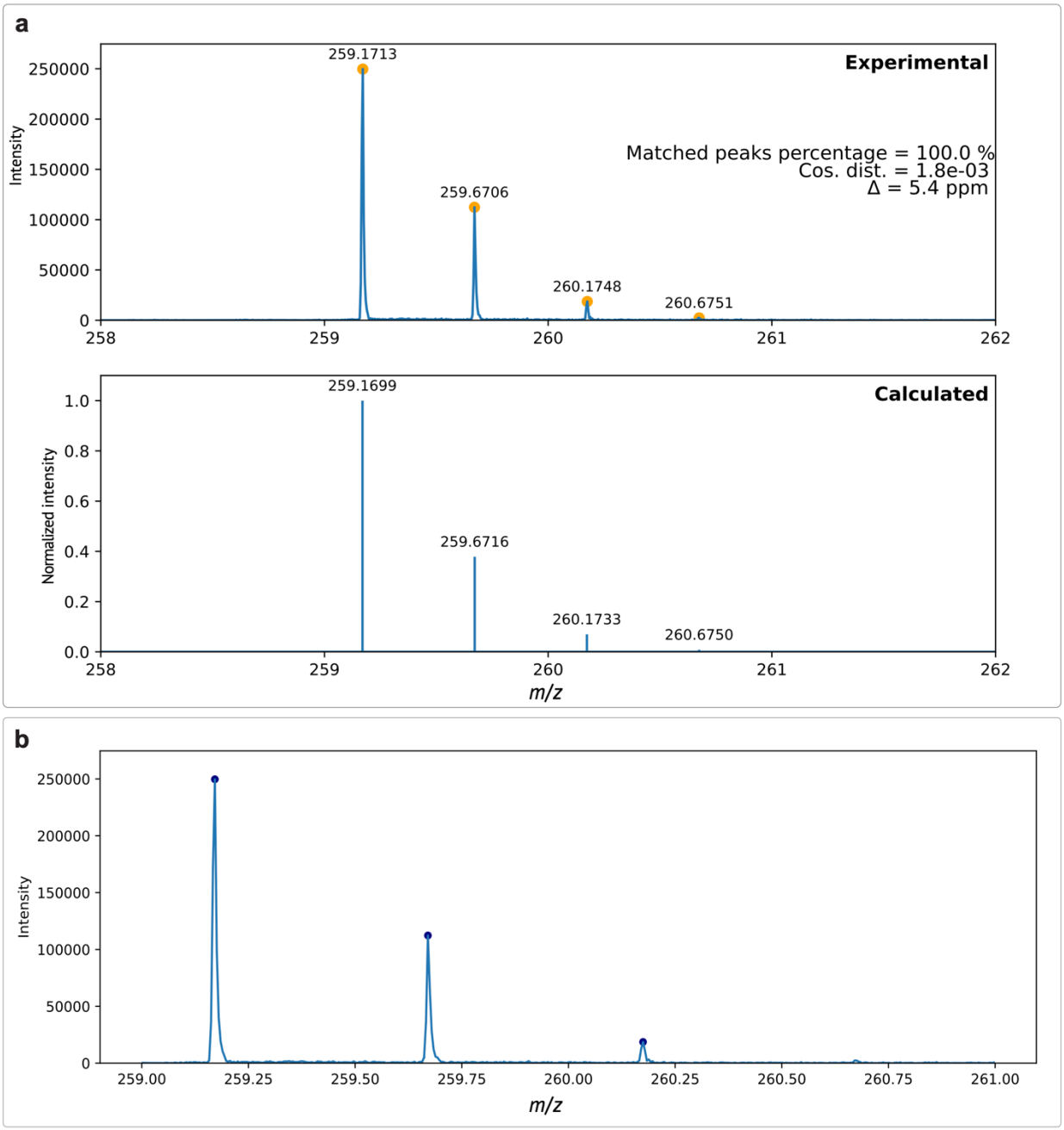
Supplementary Figure 6. Graph-based deisotoping performance on complex spectrum, which contains $C_{38}H_{47}N_3^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



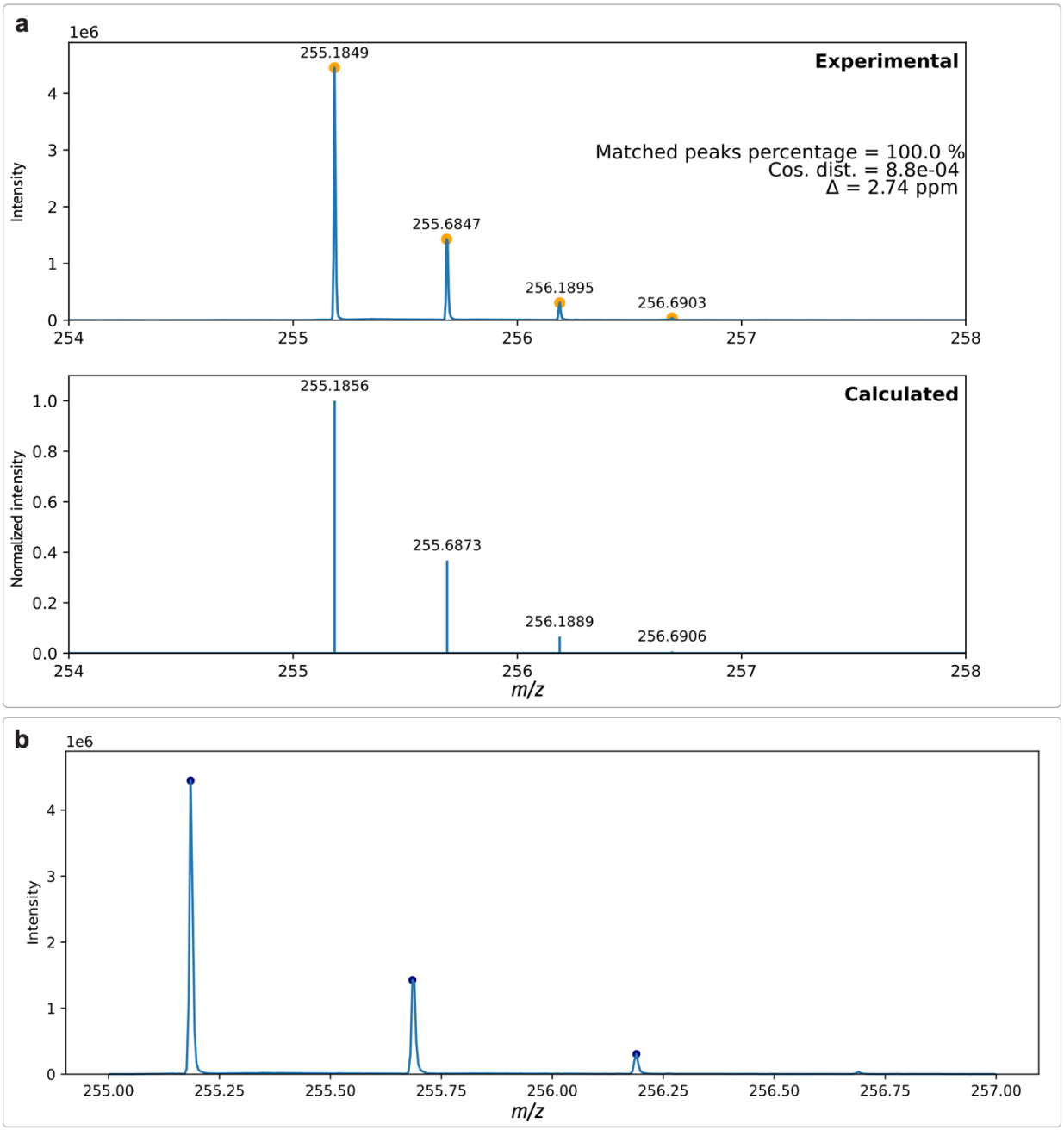
Supplementary Figure 7. Graph-based deisotoping performance on complex spectrum, which contains $C_{32}H_{35}N_3^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



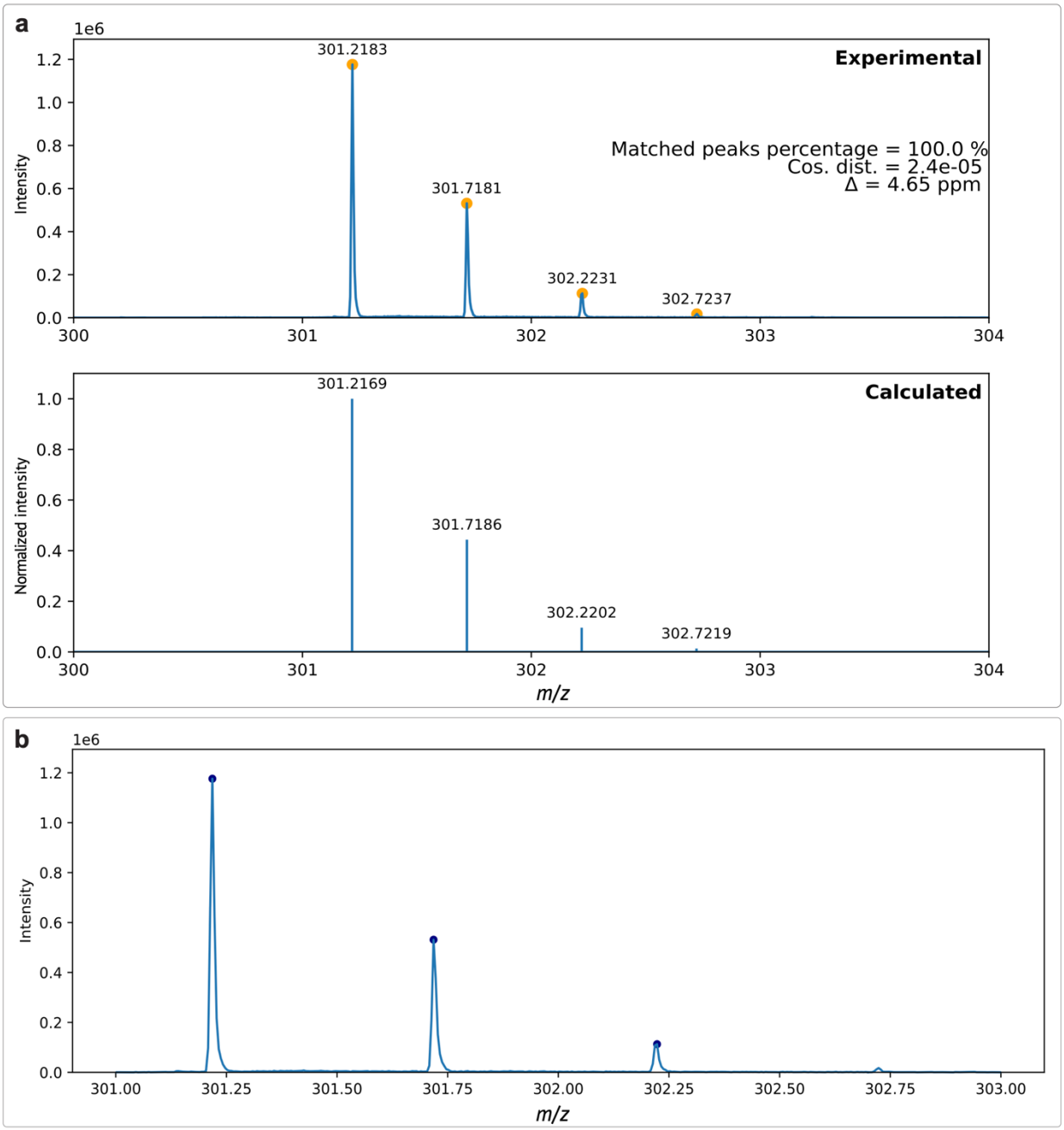
Supplementary Figure 8. Graph-based deisotoping performance on complex spectrum, which contains $C_{28}H_{34}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



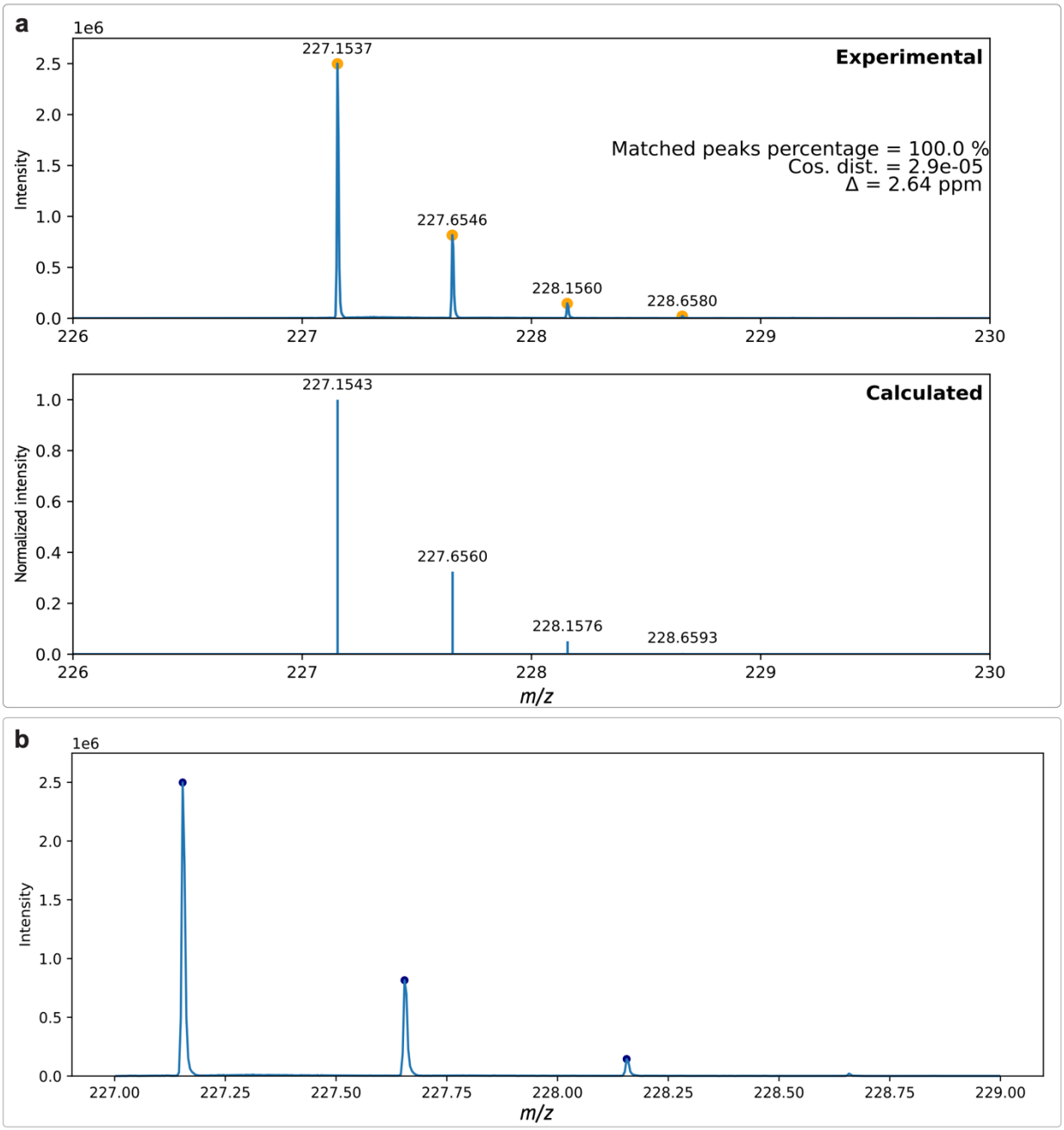
Supplementary Figure 9. Graph-based deisotoping performance on complex spectrum, which contains $C_{35}H_{42}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



Supplementary Figure 10. Graph-based deisotoping performance on complex spectrum, which contains $C_{34}H_{46}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.

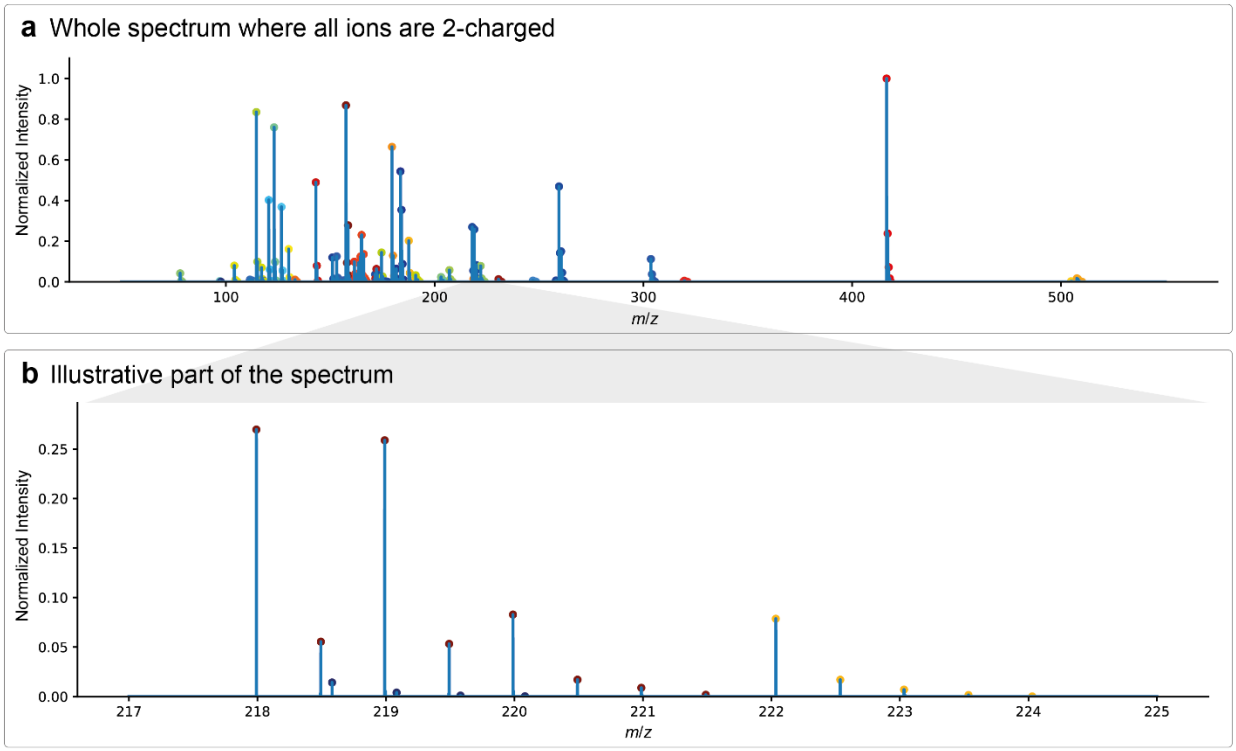


Supplementary Figure 11. Graph-based deisotoping performance on complex spectrum, which contains $C_{41}H_{54}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.



Supplementary Figure 12. Graph-based deisotoping performance on complex spectrum, which contains $C_{30}H_{38}N_4^{2+}$ ion. (a) MEDUSA compound presence verification. (b) deisotoping visualization.

Supplementary Note 1.6 Deisotoping robustness



Supplementary Figure 13. An example of deisotoping performance in a case where all ions in mass spectra are double-charged. The mass spectrum was synthetically generated with sampling molecules from the PubChem database and calculating isotopic distributions.

Supplementary Note 2 Data collection and preprocessing

Supplementary Note 2.1 Lorentzian and Full Width at Half Maximum (FWHM)

Lorentz function **(1)** is used to describe the shape of spectrum peaks. We applied Lorentzian convolution to theoretic isotopic distributions to make it more similar to experimental data. The parameter 2γ in **(1)** is FWHM – full width at half maximum of peak. This value is used to perform Lorentzian convolution correctly.

$$f(x) = \frac{1}{\pi}\left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right] \quad (1)$$

The approach to take FWHM was to predict it by linear regression model using *m*/*z* and intensities as features. Fitted equation is:

$$\mathrm{FWHM} = a * \frac{m}{z} + b * I + c \quad (2)$$

where a, b and c are parameters.

Additionally, we fitted another equation:

$$\mathrm{FWHM} = a * \sqrt{\frac{m}{z}} + b * I + c \quad (3)$$

that also performed good results. Interestingly, that the best Deep Learning models were trained on the dataset generated with the equation (2), while the best classical ML models were trained on the dataset generated with the equation (3).

The data from two experimental mass spectra with peaks in the area of *m*/*z* values from 80 to 1600 were used to train the linear model. Then, on the dataset preparation step, the prediction of FWHM for each isotopic distribution and Lorentz convolution using this value was performed.

Supplementary Note 2.2 Synthetic dataset preparation

49 104 chemical formulas were selected from the PubChem database. Among them, 1 789 contain Pd, 2 186 – Ni, 3 092 – Cu, 1 838 – Ir, 775 – Ag, 1 881 – Ru, 9 910 – Cl, 6 932 – Br, and 24 960 compounds without elements listed above. The molar mass of each selected compound was less than 1 210 g mol^{-1}. Theoretical isotopic distributions were calculated using the MEDUSA python package. *m*/*z* determination error was selected from normal distribution with the mean of 0 and standard deviation equal to 0.003. Ion charges were selected randomly (70% – single charged, 20% – double charged, 10% – triple charged).

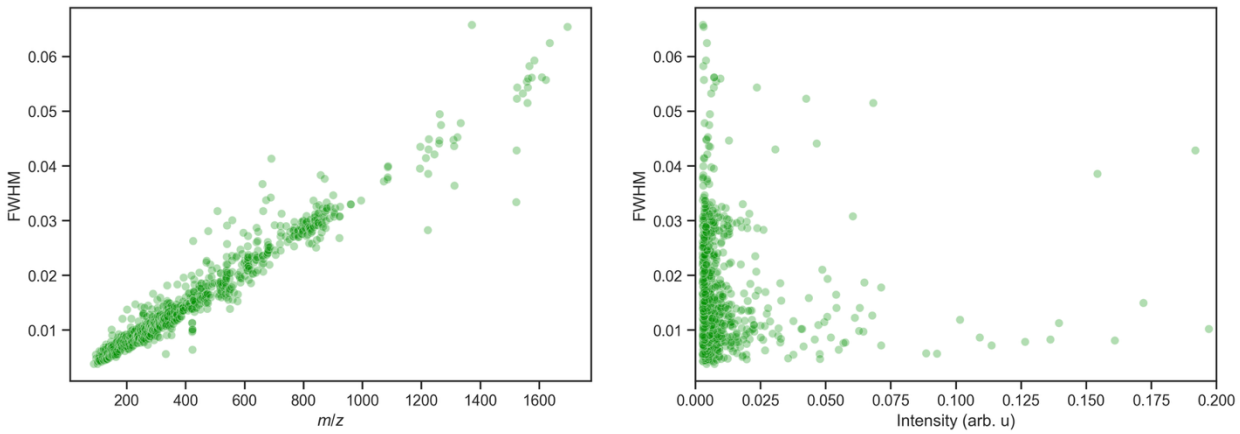
For each distribution FWHM value was predicted by a linear model trained on the data from two experimental mass spectra. Features were *m*/*z* and intensities, which values were greater than 0.0028 of the maximum intensity. Dependencies of FWHM from *m*/*z* and FWHM from intensity are presented on **Supplementary Figure 14**. Validation metrics for the linear model are presented on **Supplementary Table 1**.

Supplementary Table 1. Regression metrics for the model, which predicts FWHM from experimental data.

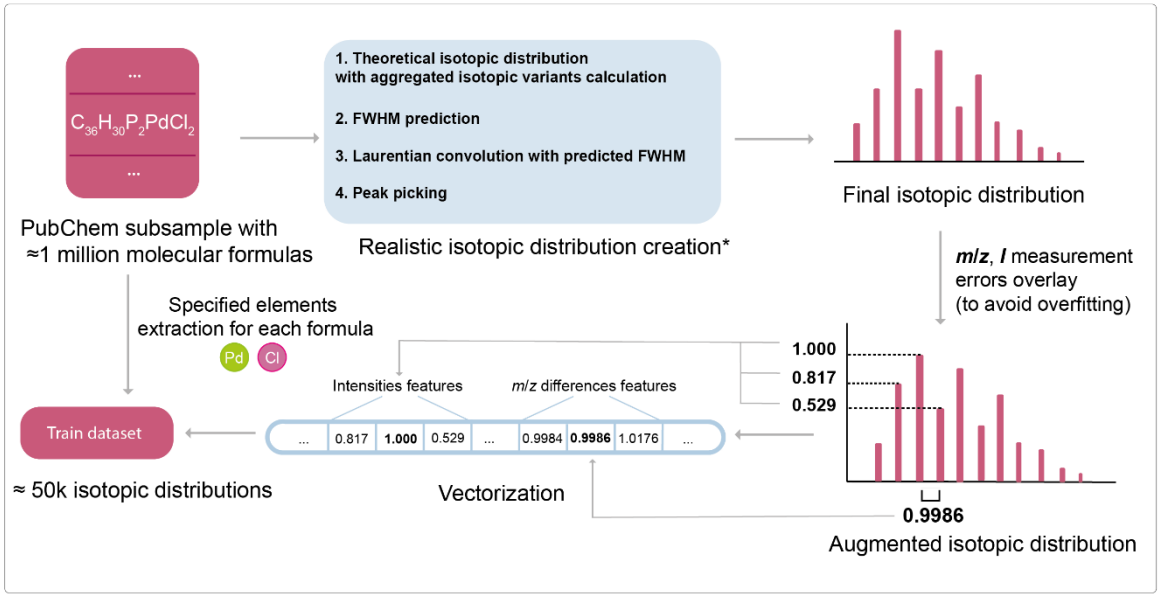
Metric	R^2	MSE	MAE	MAPE
Value	0.9355	$5.565 * 10^{-6}$	$1.473 * 10^{-3}$	0.1064

Lorentzian convolution was performed using this value. After that, spectra were deconvoluted, yielding new *m*/*z* and intensity values. The intensity vector with a length of 25 was centered with the corresponding $\Delta m/z$ values vector with a length of 25. These vectors were concatenated, giving a final feature vector with the size of 50. This vector, vectors of *m*/*z*, and element labels were added to the final dataset **(Supplementary Figure 15)**.

There is an option to add *m*/*z* values to the classifier`s input. *m*/*z* values are divided by 1000 to scale the data and concatenated to the input vector.



Supplementary Figure 14. Experimental data distribution for FWHM (full width at half maximum of peak) model prediction, where only data points with intensity > 0.0028 of maximal intensity were taken. Source data are provided as a Source Data file.



Supplementary Figure 15. Dataset generation procedure. FWHM – Full Width at Half Maximum of peak.

Supplementary Note 2.3 Extended synthetic dataset

To increase the model`s precision we enriched our dataset with new compounds. 3000 organic compounds containing at least 1 carbon and 2 hydrogen atoms were added. Various manipulations were performed on each compound as follows:

$C_nH_mR \rightarrow C_nH_{m-2}RCl$

$C_nH_mR \rightarrow C_nH_{m-1}RPd$

$C_nH_mR \rightarrow C_nH_mRPdCl$

$C_nH_mR \rightarrow C_nH_mRPdBr$

$C_nH_mR \rightarrow C_nH_mRPd_2$

Here, C – carbon, H – hydrogen, Pd – palladium, Cl – chlorine, Br – bromine, R – other elements such as oxygen, nitrogen, sulfur, or phosphorus.

Consequently, the synthetic dataset was augmented by 18 000 new compounds.
Extended dataset was used exclusively to train double-encoder neural network model with augmentations.

Supplementary Note 2.4 Experimental dataset statistics

Testing dataset was collected from manually labelled 471 isotopic distributions obtained from experimental mass spectra. Among these distributions, 39 contain Pd, 23 – Ni, 7 – Cu, 7 – Ag, 68 – Cl, 39 – Br, and 330 compounds without elements listed above. All samples containing target elements are single charged. 14 compounds without target elements are double charged.

Supplementary Note 3 Elemental composition prediction methods

Supplementary Note 3.1 Implementation details of the classical ML approach

Cross-validation of classical machine learning models was performed using GridSearchCV (from Scikit Learn) to find the best parameters and using the Multilabel Stratified KFold implementation (from iterative-stratification Python package), dividing the training dataset into 5 parts and using ROC AUC as the key metric. The hyperparameters that were cross-validated and the best model values are presented in **Supplementary Table 2.**

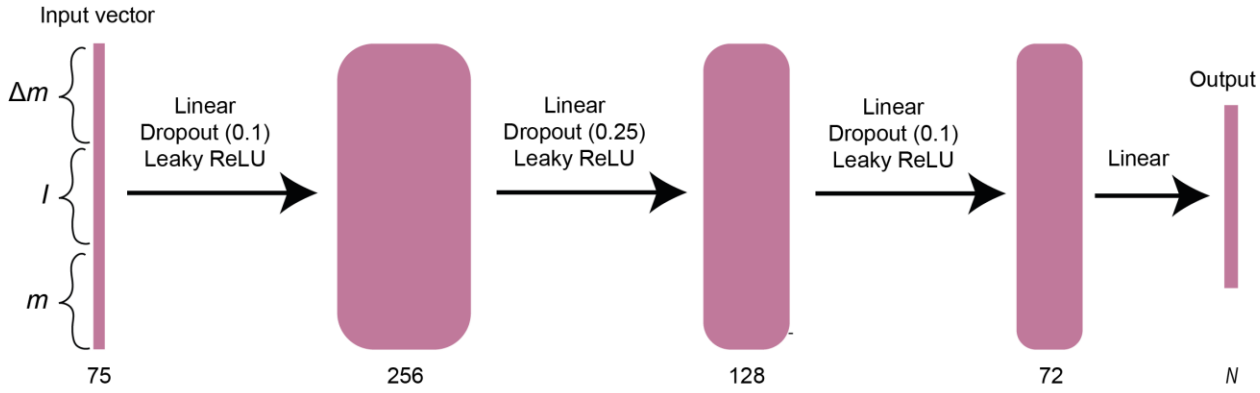
Supplementary Table 2. Hyperparameters and best scores. The hyperparameters of the best model are in bold.

Model	Hyperparameters	Best score
XGBoost	n_estimators: **300**, 500, 1000, 1500	0.9988
	max_depth: 6, 8, **10**, 12	
Support Vector Classifier	C: 10^{-4}, 10^{-3}, 10^{-2}, 0.1, **1**	0.9739
	kernel: linear, poly, **rbf**	
Random Forest Classifier	n_estimators: 50, 100, 300, 500, 700, **900**	0.9981
	max_depth: 4, 6, 8, **12**	
CatBoost	n_estimators: 300, 500, 1000, **1500**	0.9989
	max_depth: 6, 8, 10, **12**	
Multi-layer Perceptron	solver: **sgd**, adam	0.9953
	learning rate: constant, **adaptive**	
	activation: logistic, tanh, **relu**	
	learning rate init: 10^{-5},10^{-4}, 10^{-3}, 10^{-2}, **0.1**	

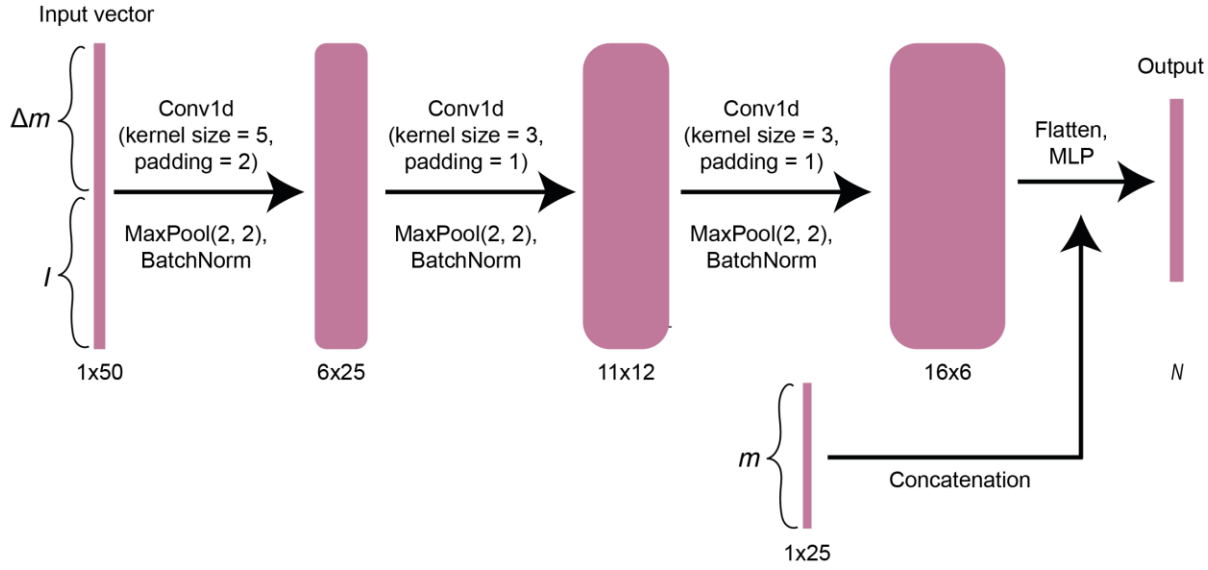
Supplementary Note 3.2 Neural network architectures

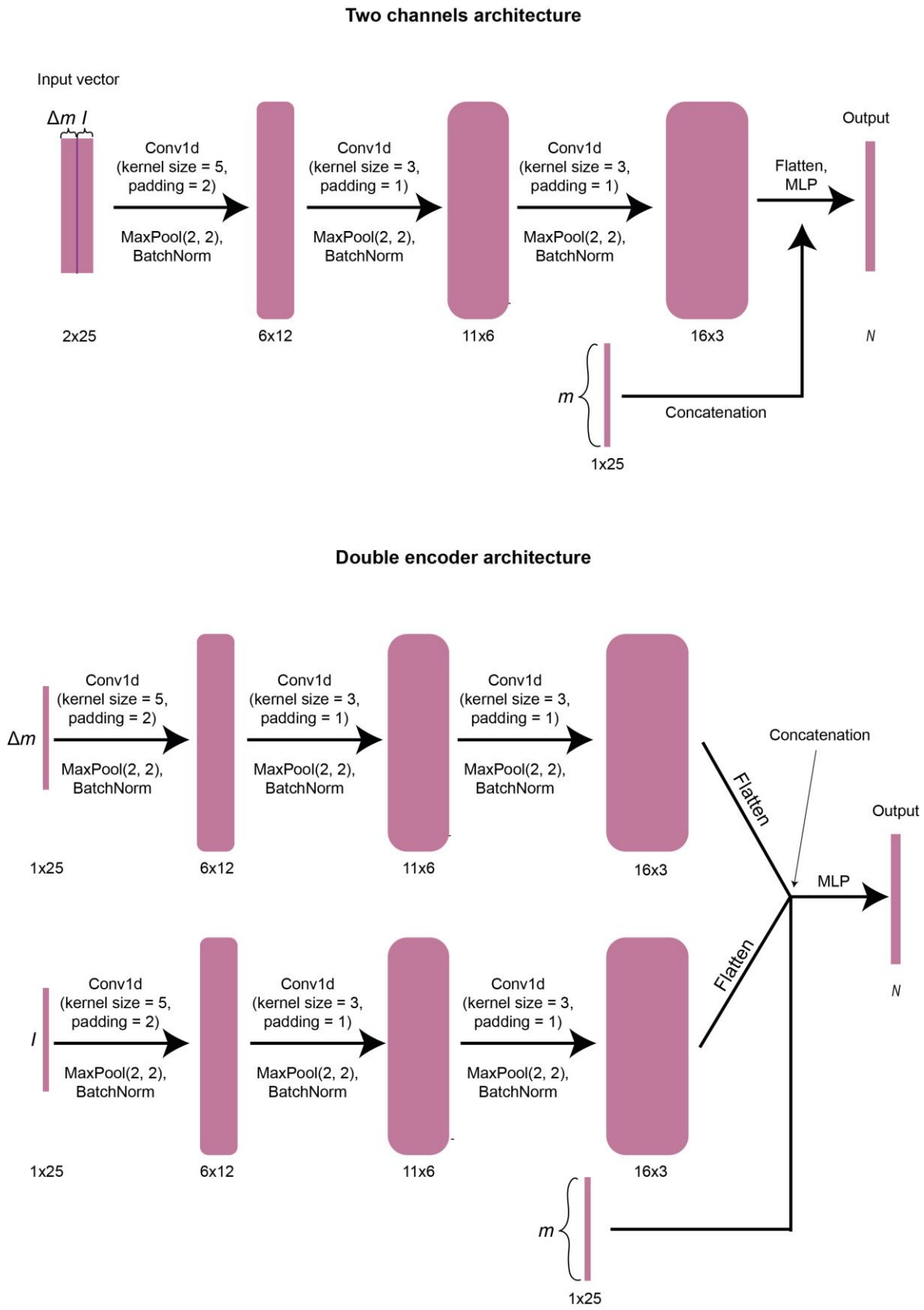
Here, the schemes of final models` architectures are presented **(Supplementary Figure 16)**. The MLP classifier in the convolutional networks has the same structure as in the fully-connected architecture. Classifier takes convolutional encoder`s output that is flattened and concatenated with the vector of m/z. N is the number of elements to be predicted. Final model's hyperparameters are specified in the **Supplementary Table 3.**

Fully-connected architecture



Single encoder architecture





Supplementary Figure 16. Four neural network architectures.

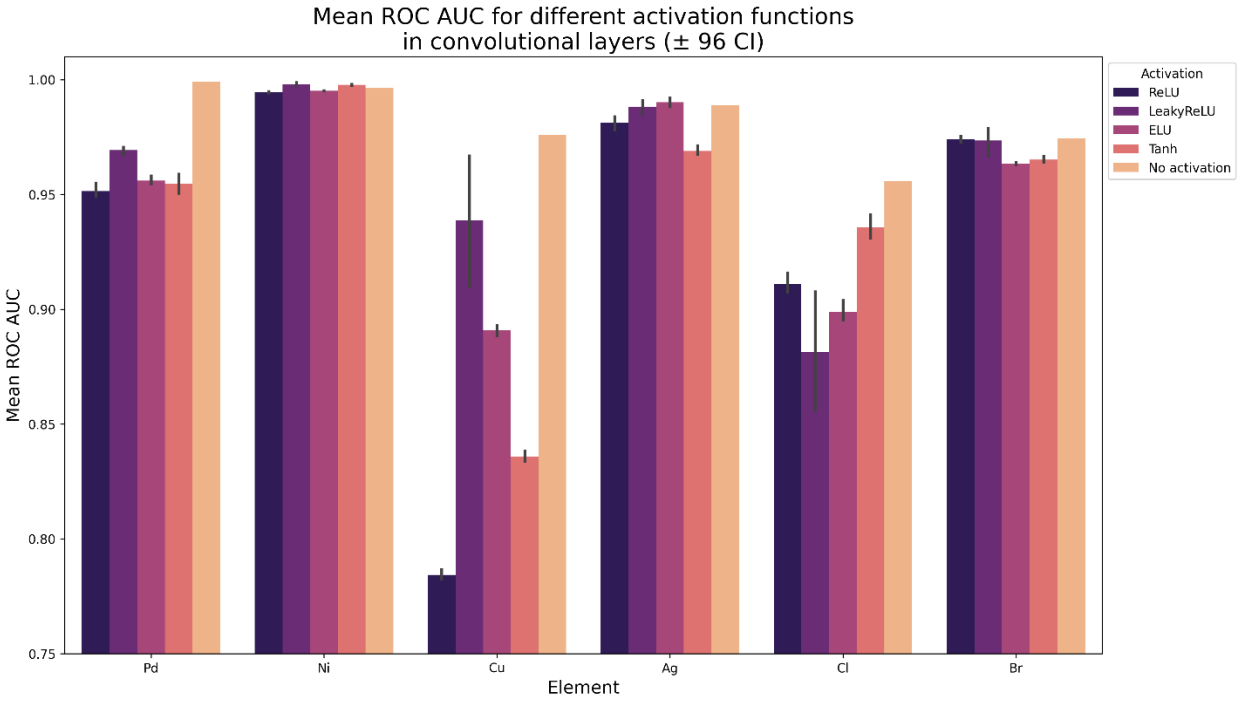
Supplementary Note 3.3 Neural network hyperparameters

Supplementary Table 3. Hyperparameters of the final neural network.

Model	**MLP**	**Single encoder convolutional model**	**Two channels convolutional model**	**Double encoder convolutional model**
Activation function	Leaky ReLU	Leaky ReLU	Leaky ReLU	Leaky ReLU
Optimizer	Adam	Adam	AdamW	AdamW
Learning rate	0.0012	0.0012	0.001	0.0012
Learning rate scheduler	StepLR (step_size=8, gamma=0.5)	StepLR (step_size=8, gamma=0.5)	StepLR (step_size=10, gamma=0.5)	StepLR (step_size=8, gamma=0.5)
Loss function	Binary cross entropy	Binary cross entropy	Binary cross entropy	Binary cross entropy

Supplementary Note 3.4 Activation function in convolutional layers.

It should be noted that the convolution layers in the final architectures do not incorporate activation functions. The final model (comprising two encoders) was trained using various activation functions (ReLU, LeakyReLU, ELU, Tanh), and the outcomes were compared. Our observations revealed that the model exhibiting the highest ROC AUC values on the experimental data was the one with no activation in the convolution layers, denoted as "No activation" in the diagrams. **Supplementary Figure 17** displays mean test ROC AUCs of 5 training runs for each mentioned activation function. Error-bars show 96% confidence interval.



Supplementary Figure 17. Comparison of experimental ROC AUC scores for models with different activation functions in the convolution layers. Source data are provided as a Source Data file.

Supplementary Note 3.5 Data augmentations

In case of low compound`s concentration it`s peaks on the mass spectrum could be approximately the same size as noise peaks. It could lead to incorrect performance of deisotoping and peak-picking algorithms and, as a result, to inaccurate model`s predictions. To struggle with this issue and increase model`s robustness we performed some data augmentations.

For each peak within every isotopic distribution, we computed the probability of peak replacement using the following equation:

$$p = \frac{1 - i^{\alpha}}{\beta} \quad (4)$$

where i – relative intensity, α and β – some real numbers.

Subsequently, a Bernoulli random variable B ~ Bernoulli(p) was sampled. If B = 1, the peak intensity was replaced by a value sampled uniformly from $[0, 10^{-3}]$; otherwise, the original intensity was retained. The augmentation algorithm is delineated in pseudocode.

FOR i IN intensities:

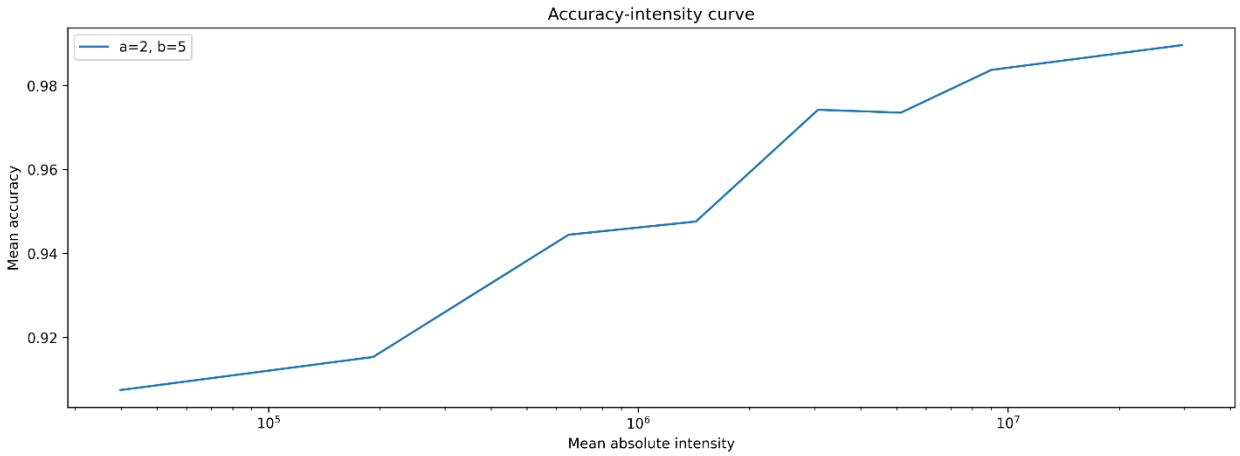
$$p <- \frac{1-i^{\alpha}}{\beta}$$

$$\text{to\_replace} <- \text{Bernoulli}(p)$$

IF to_replace $= 1$:

$$i <- \text{Uniform}(0, 10^{-3})$$

To estimate the augmentations` quality we used AUROC and a specific metric that we call “Accuracy AUC”. To evaluate it we at first splitted experimental spectra into 8 groups by absolute intensity values. These groups lie in intervals 10^3–10^5, 10^5–$10^{5.5}$, $10^{5.5}$–10^6, 10^6–$10^{6.3}$, $10^{6.3}$–$10^{6.6}$, $10^{6.6}$–$10^{6.8}$, $10^{6.8}$–$10^{7.1}$, $10^{7.1}$–10^9. After that, mean absolute intensity and mean model`s accuracy were calculated in every interval. As a result, we obtained a curve **(Supplementary Figure 18)**. To evaluate the final metric, we calculated the area under this curve and divided it to the difference between the highest and lowest intensity points. So, the result value lies between 0 and 1 and is expected to represent the prediction quality with respect to the absolute intensity.



Supplementary Figure 18. The dependence of accuracy on the absolute intensity. Source data are provided as a Source Data file.

Supplementary Note 3.6 Models trained on the augmented data

To select the optimal parameters α and β and evaluate augmentation`s quality we conducted several experiments.

For each pair (α, β), where α in (1, 2, 3, 4, 5) and β in (4, 5, 6, 7, 8, 9, 10), we trained five models with double-encoder architecture and calculated mean “accuracy AUC” and mean AUROCs for each target element. The results are shown below **(Supplementary Figure 19)**.

As final parameters (α, β) = (2, 5) were chosen. Metrics for this model trained on the extended dataset are shown in **Supplementary Table 4**. Hyperparameters of the final model are shown in **Supplementary Table 5.**

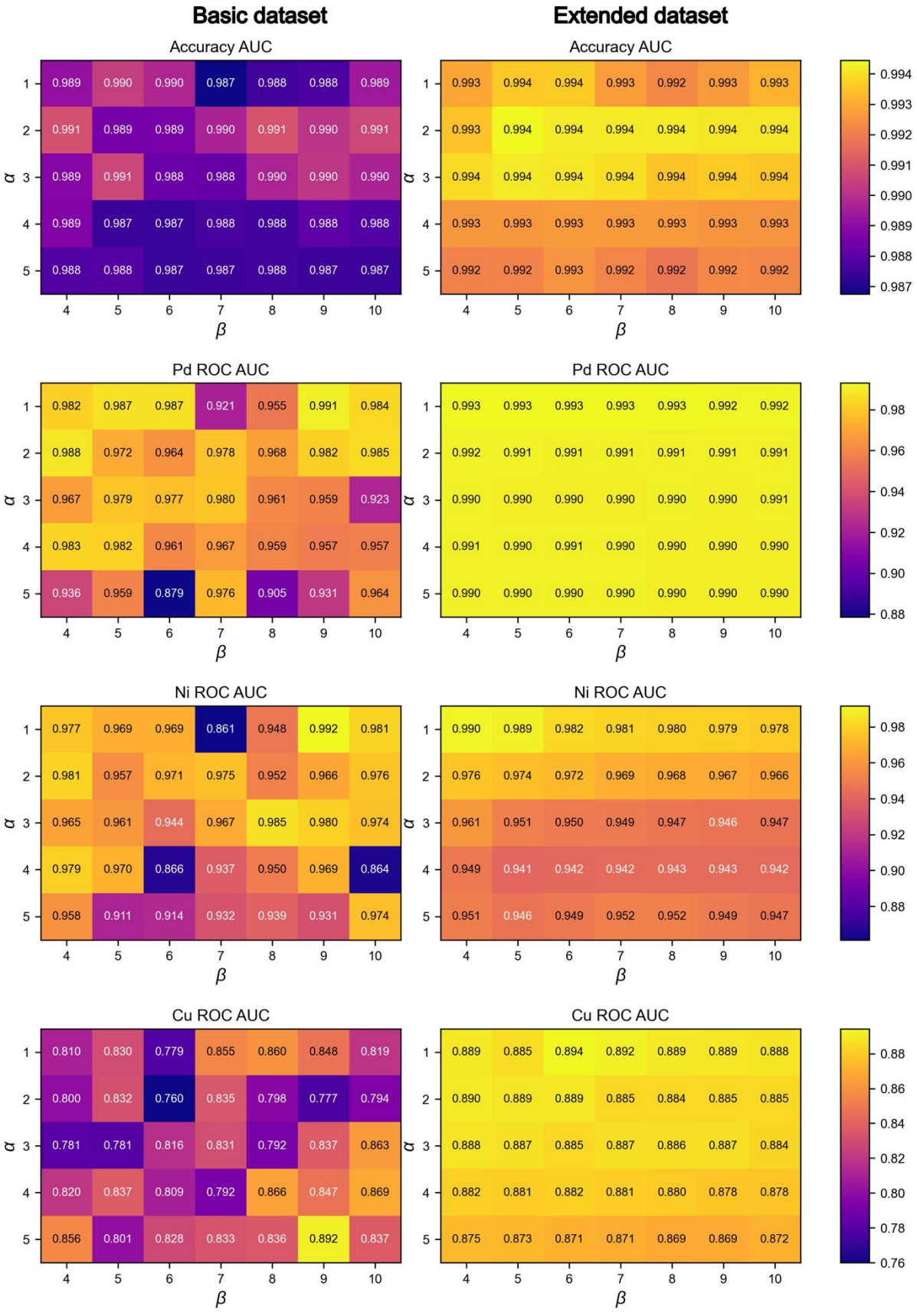
While training the augmented neural networks we observed that models have large variance. To fix that we decreased the learning rate to 10^{-5} **(Supplementary Figure 20)** and set a step learning rate scheduler with step size 8 and learning rate decay parameter 0.4 **(Supplementary Figure 21)**. But that led to lessening of the prediction threshold (if prediction probability is greater than threshold, it rounds to class 1, and to class 0 otherwise). So, we found thresholds for each element to maximize the F1-score. Thresholds were selected using experimental (test) dataset, because it has a different distribution than synthetic data.

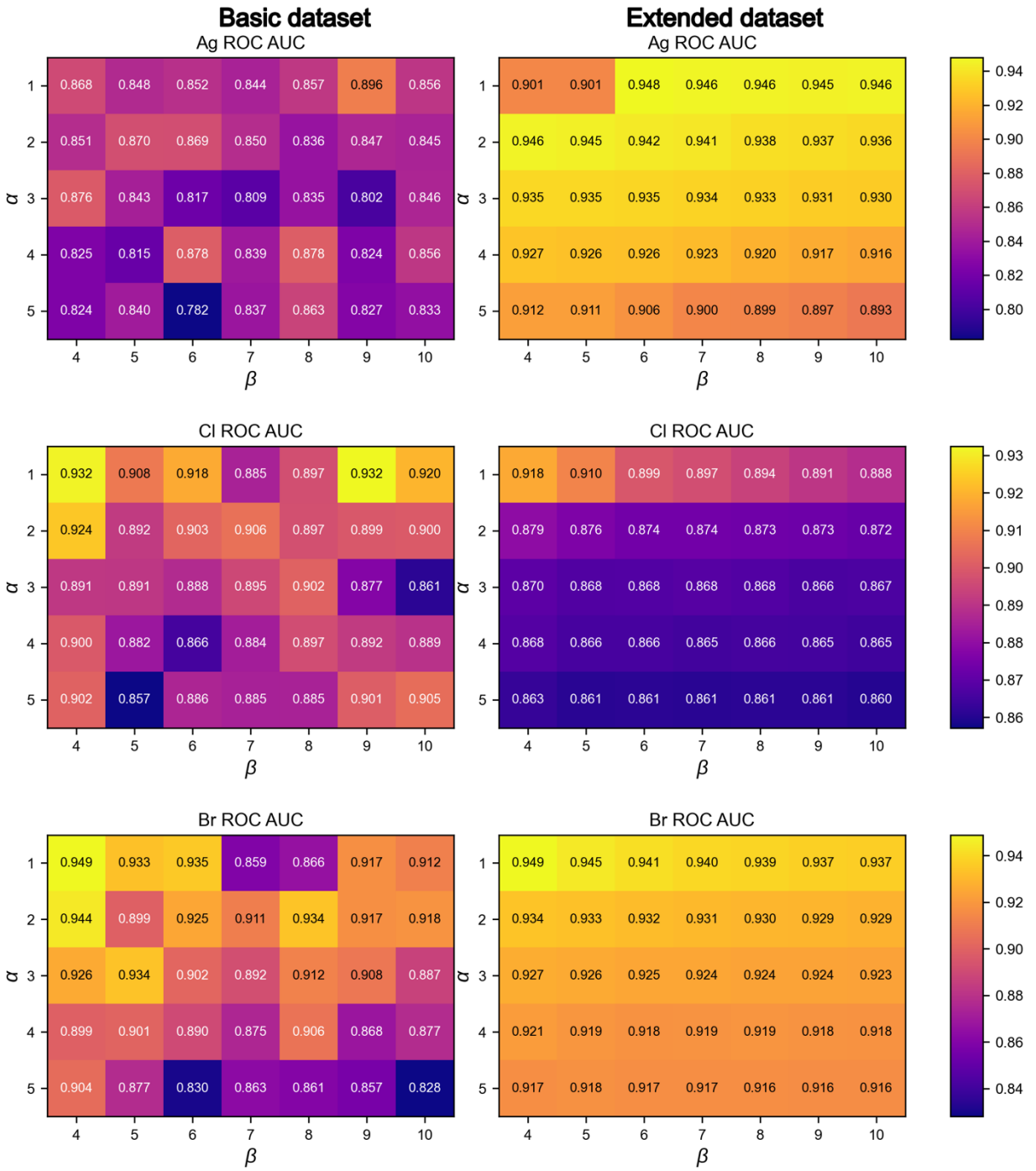
Supplementary Table 4. Prediction threshold, accuracy, precision, recall and ROC AUC scores calculated on the experimental data for the model with augmentations.

Element	Threshold	Accuracy	Precision	Recall	ROC AUC
Pd	0.4375	0.9533	0.6545	0.9231	0.9719
Ni	0.1539	0.9682	0.6250	0.8696	0.9841
Cu	0.1355	0.8535	0.0811	0.8571	0.7749
Ag	0.0874	0.9512	0.233	1.000	0.9529
Cl	0.2852	0.7856	0.3907	0.8676	0.8612
Br	0.4831	0.9597	0.8125	0.6667	0.9252

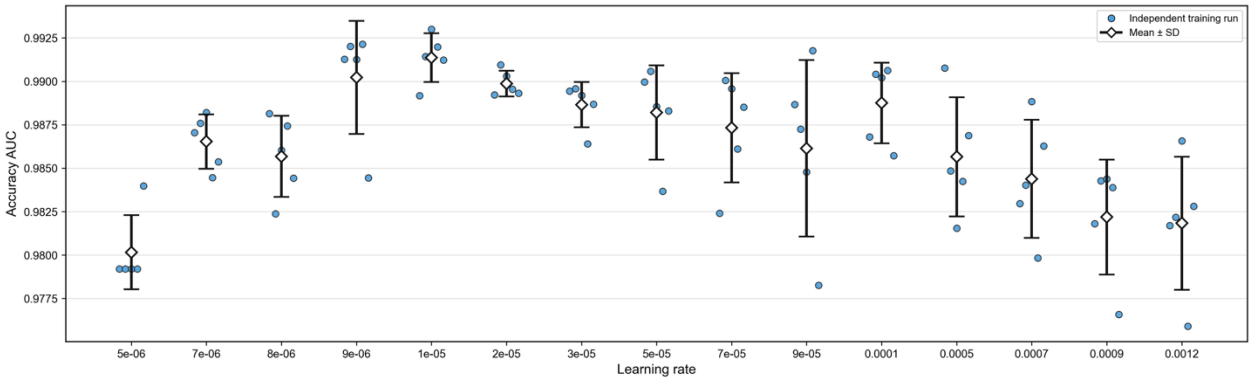
Supplementary Table 5. Hyperparameters of the model, trained on the augmented data.

Model architecture	α	β	Activation function	Optimizer	Learning rate	Learning rate scheduler	Loss function
Double-encoder	2	5	LeakyReLU	AdamW	10^{-5}	StepLR (step_size=8, gamma=0.4)	Binary Cross Entropy

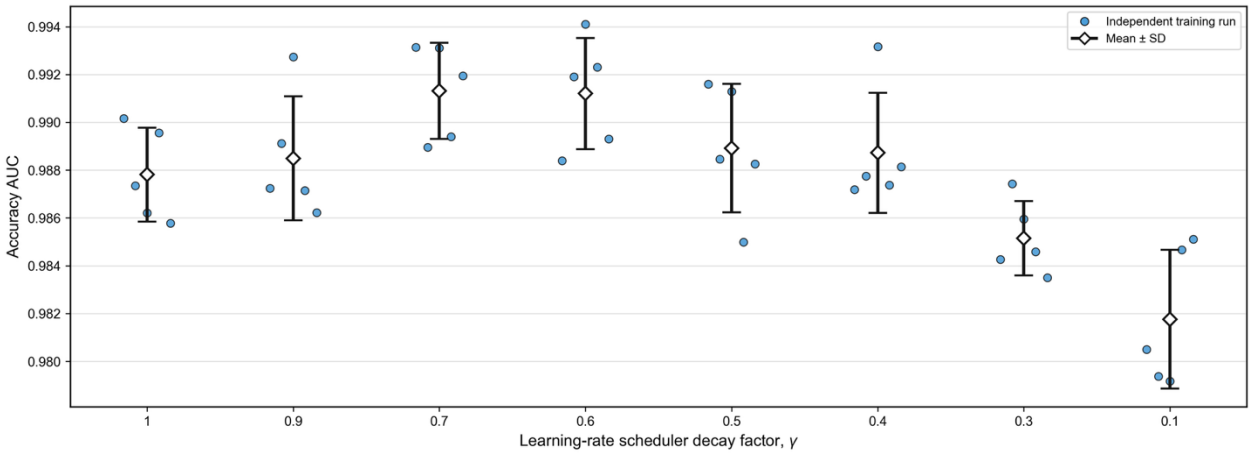

Basic dataset
Extended dataset
Accuracy AUC
Pd ROC AUC
Ni ROC AUC
Cu ROC AUC
α
β



Supplementary Figure 19. Heatmaps with the metrics (accuracy AUC and elementwise ROC AUC scores) of the models with different augmentation parameters α and β on the experimental data. Results are shown for the basic (left column) and augmented (right column) datasets.



Supplementary Figure 20. Accuracy AUC scores of five independently trained models without augmentations evaluated using different learning-rate parameters. Blue circles indicate the score from each independent model-training run (n = 5 per learning-rate setting); all runs used the same fixed data split. Diamonds and error bars show the mean ± standard deviation (SD). A learning rate of 10^{-5} was chosen as the optimal value. Accuracy AUC scores were calculated using the experimental data. The statistical significance of the overall difference among learning-rate settings was evaluated using a two-sided Kruskal–Wallis test ($H(14) = 45.63$, $P = 3.22 \times 10^{-5}$). This omnibus result indicates that at least one setting differs; no pairwise significance is inferred. Source data are provided as a Source Data file.



Supplementary Figure 21. Accuracy AUC scores of five independently trained models without augmentations evaluated using different learning-rate decay parameters (γ). Blue circles indicate the score from each independent model-training run (n = 5 per γ setting); all runs used the same fixed data split. Diamonds and error bars show the mean ± standard deviation (SD). A γ value of 0.4 was chosen as the optimal value (0.4 on the figure). Accuracy AUC scores were calculated using the experimental data, with the learning rate set to 10^{-5}. The statistical significance of the overall difference among γ settings was evaluated using a two-sided Kruskal–Wallis test ($H(7) = 24.52$, $P = 9.23 \times 10^{-4}$). This omnibus result indicates that at least one setting differs; no pairwise significance is inferred. Source data are provided as a Source Data file.

Supplementary Note 3.7 Statistical analysis of multiple spectra

General procedure:

1. Whole spectrum is divided to the isotopic distributions by our deisotoping algorithm.
2. The list of distributions is filtered:
    a. Distributions with the number of peaks less than 5 are removed.
    b. Distributions that start with their most intensive peak are removed (because of the specific Pd distribution pattern)
3. For each remaining distribution the probability score of Pd content is obtained by our neural network.
4. Statistics are calculated based on the Pd content scores and intensities of the distributions.

Statistical calculations description.

- Sum. Sum of all scores of Pd content over the spectrum.
- Mean. Average of all scores of Pd content over the spectrum.
- Max. Maximum of all scores of Pd content over the spectrum.
- 95-percentile. 95-th percentile of all scores of Pd content over the spectrum.
- Count N. Count of distributions in the spectrum with Pd content score exceeding N%.
- Count N normalized. As “count N” but divided by the whole number of distributions in the spectrum.
- Weighted intensities. The sum of the summed relative intensities in the distributions multiplied by Pd content scores

$$\mathrm{WI} = \sum_{d \in distributions} \left(P_d[\mathrm{Pd}] \times \sum_{p_d \in \mathrm{peaks}} (\mathrm{Intensity})_{p_d} \right) \quad (5)$$

- Absolute weighted intensities

$$\mathrm{AWI} = \sum_{d \in \mathrm{distributions}} \left(I[P_d[\mathrm{Pd}] > 0.25] \times \sum_{p_d \in \mathrm{peaks}} (\mathrm{Intensity})_{p_d} \right) \quad (6)$$

  where *I* is an indicator.

- Inversed weighted intensity

$$\mathrm{IWI} = \sum_{d \in \mathrm{distributions}} \left(P_d[\mathrm{Pd}] \times \left(1 - \sum_{p_d \in \mathrm{peaks}} (\mathrm{Intensity})_{p_d} \right) \right) \quad (7)$$

- Normalized weighted intensity

$$\mathrm{NWI} = \frac{1}{D} \sum_{d \in \mathrm{distributions}} \left(P_d[\mathrm{Pd}] \times \sum_{p_d \in \mathrm{peaks}} (\mathrm{Intensity})_{p_d} \right) \quad (8)$$

$$where\ D\ is\ the\ number\ of\ distributions\ in\ the\ spectrum$$

- Normalized absolute weighted intensity

$$\mathrm{NAWI} = \frac{1}{D} \sum_{d \in \mathrm{distributions}} \left(I[P_d[\mathrm{Pd}] > 0.25] \times \sum_{p_d \in \mathrm{peaks}} (\mathrm{Intensity})_{p_d} \right) \quad (9)$$

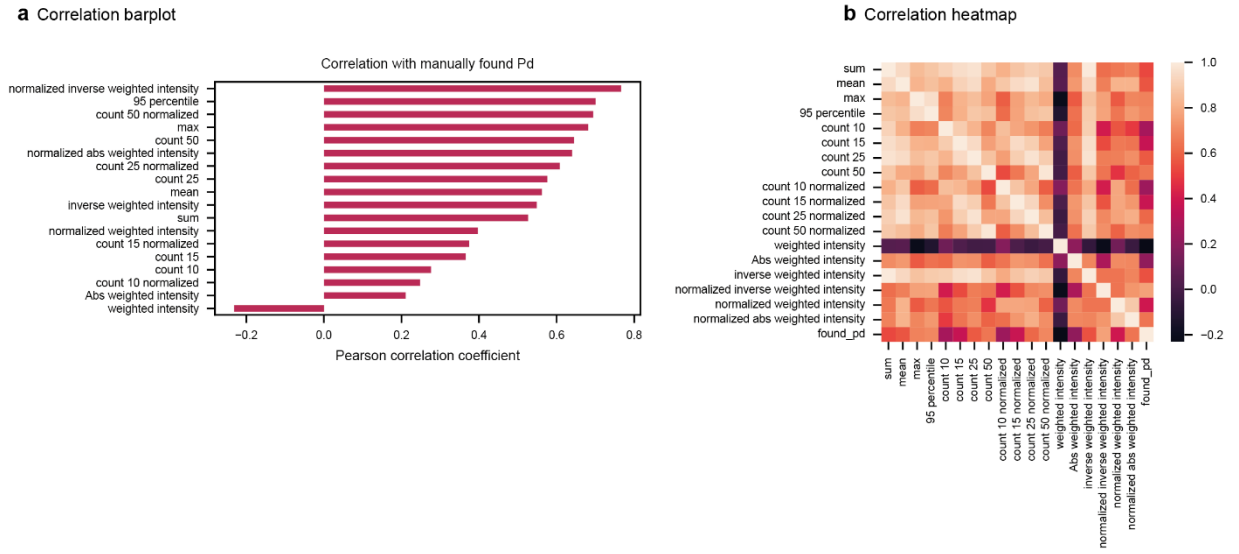
- Normalized inversed weighted intensity

$$\mathrm{NIWI} = \frac{1}{D} \sum_{d \in \text{distributions}} \left(P_d[\mathrm{Pd}] \times \left(1 - \sum_{p_d \in \text{peaks}} (\mathrm{Intensity})_{p_d} \right) \right) \quad (10)$$

For understanding the relevance of these metrics, we labelled as “1” the spectra with manually found Pd, and as “-1” the spectra without any Pd content. Additionally, we labelled as “2” the spectrum of triphenylphosphine because of exceptional number of Pd distributions. Next, we calculated the Pearson correlation between all the metrics and actual Pd content.

The correlation coefficients for the metrics and Pd content are presented in **Supplementary Figure 22a**. As shown, the “Normalized Inversed Weighted Intensity” (NIWI) demonstrates the highest correlation with Pd content. However, this metric was excluded from the main article due to its limited interpretability. The observed high correlation may be attributed to the fact that most Pd distributions in the spectra exhibit low intensities. In contrast, for spectra where Pd is associated with prominent peaks, the NIWI metric may yield misleading results.

Additionally, pairwise correlation coefficients for all metrics are depicted as a heatmap in **Supplementary Figure 22b**. It is worth noting that “Weighted intensity” (WI) has a negative (or almost negative) correlation with all other metrics. It is also of interest that normalization of WI improves its correlation.



Supplementary Figure 22. Correlation Analysis of Metrics with Manually Identified Palladium Content. (a) Bar plot of Pearson correlation coefficients between metrics and manually identified Pd content. (b) Heatmap of pairwise Pearson correlation coefficients between all metrics. Source data are provided as a Source Data file.

Supplementary Note 3.8 New element onboarding

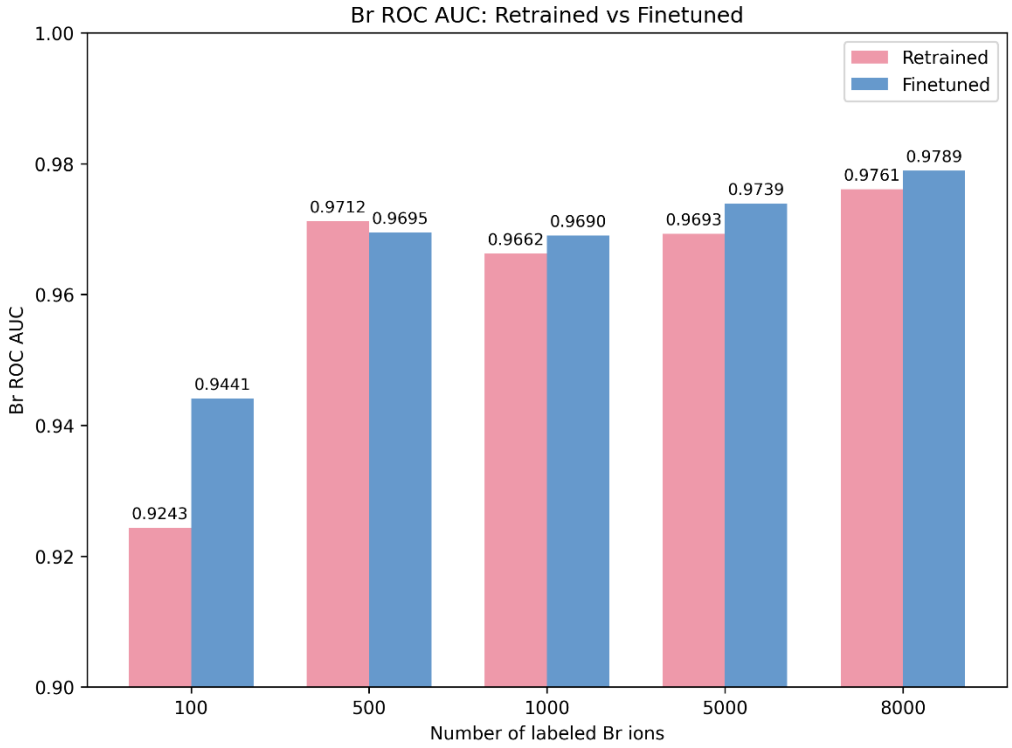
Our framework allows new target elements addition. It could be done through retraining the whole model or finetuning of existing model trained on lesser number of elements.

Also, it is important to determine necessary number of ions with new element in the training set.

To demonstrate the new element onboarding process, we trained our double encoder model from scratch on the training dataset with excluded all bromine (Br) containing distributions and next, we added Br as a new element. Especially:

1. We trained our double-encoder model from scratch on the original training dataset, excluding all labeled Br distributions.
2. We then added 100, 500, 1,000, 5,000, and 8,000 Br samples to the original dataset. For each case, we fine-tuned the pretrained double-encoder model (with frozen convolutional layers) and, for comparison, also trained a model from scratch.
3. We took 100 epochs for retraining and 50 epochs for finetuning. Learning rate was $2*10^{-5}$ for retraining and $1*10^{-5}$ for finetuning.
4. Final performance was evaluated on the experimental test set.

Supplementary Figure 23 demonstrates, that satisfactory quality reaches already with 500 labeled Br distributions. Also, it is shown, that in the most cases finetuning gives better results than retraining.
Detailed new element onboarding instruction and corresponding code is located in our GitHub repository in `*research/new_element_onboarding*`.



Supplementary Figure 23. Bromine (Br) ROC AUC evaluated on the experimental dataset depending on number of training Br ions.

We also observed that adding a new element generally improves prediction quality for the original elements. To evaluate this effect, we fixed results with 8 000 Br ions and ranked the results for other elements **(Supplementary Table 6)**.

Supplementary Table 6. Test ROC AUC values for target elements obtained using a model trained on data without Br, and models either retrained or finetuned on data with 8,000 Br ions added. The best results for each element are highlighted in bold, and the corresponding rank is shown in parentheses.

	Without Br	Retrain	Finetune
Pd	0.9851 (2)	**0.9863 (1)**	**0.9863 (1)**
Ni	0.9923 (2)	0.9902 (3)	**0.9942 (1)**
Cu	0.8881 (3)	**0.9219 (1)**	0.8946 (2)
Ag	**0.9632 (1)**	0.9362 (2)	0.9352 (3)
Cl	0.9653 (3)	**0.9707 (1)**	0.9683 (2)
Average rank	2.2	**1.6**	1.8
Average ROC AUC	0.9588	**0.96106**	0.95572

As **Supplementary Table 6** shows, for all cases except silver the performance increases after Br onboarding.

Supplementary Note 4 Model experimental validation

Supplementary Note 4.1 Neural network quality comparison

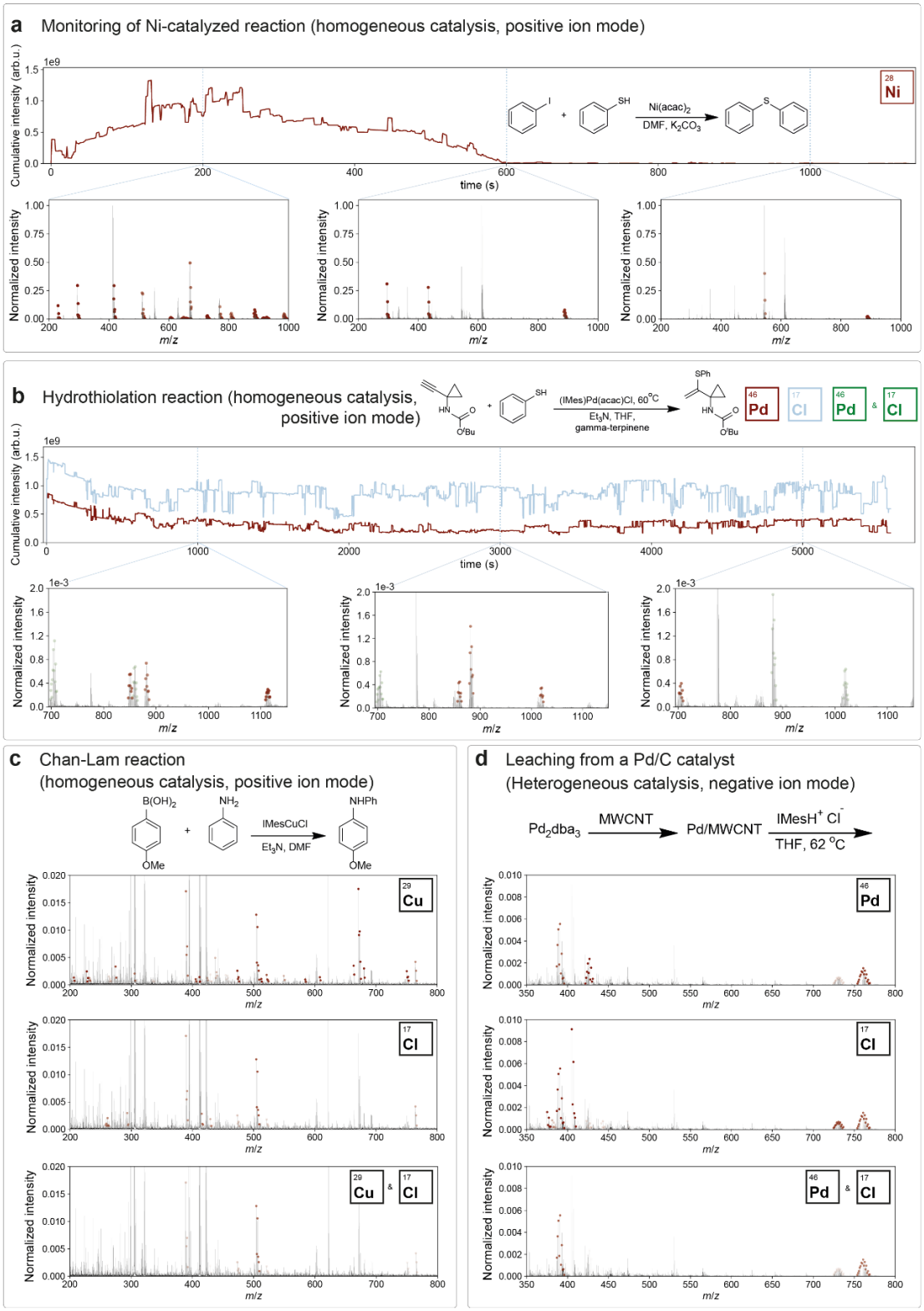
Test classification metrics obtained for each proposed neural network are presented in the **Supplementary Table 7.** The best metric values across specified element are highlighted in bold.

Supplementary Table 7. Accuracy, precision, recall and ROC AUC values on experimental data of all NN models for each element. Best results are highlighted in bold.

Model	Element	Accuracy	Precision	Recall	ROC AUC
MLP	Pd	0.9469	**1.0000**	0.3590	0.9640
	Ni	0.9979	**1.0000**	0.9565	0.9985
	Cu	**0.9639**	**0.2500**	0.7143	0.8427
	Ag	0.9851	0.0000	0.0000	0.9754
	Cl	**0.9299**	**0.8889**	0.5882	**0.9668**
	Br	**0.9682**	**0.8929**	**0.6757**	0.9704
	Average	**0.9653**	0.6720	0.5489	0.9530
	Pd	0.9597	**1.0000**	0.5128	0.9839
	Ni	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Cu	0.9448	0.1481	0.5714	0.9012

Single encoder convolutional model	Ag	0.9745	0.3077	**0.5714**	0.9809
	Cl	0.8790	0.5591	**0.7647**	0.9079
	Br	0.9490	0.8095	0.4595	0.9707
	Average	0.9512	0.6374	0.6466	0.9574
Two channels convolutional model	Pd	0.9597	**1.0000**	0.5128	0.9859
	Ni	0.9936	0.9167	0.9565	0.9969
	Cu	0.9533	0.1429	0.4286	0.8522
	Ag	**0.9894**	**1.0000**	0.2857	0.9683
	Cl	0.8641	0.5270	0.5735	0.9320
	Br	0.9469	0.7500	0.4865	**0.9758**
	Average	0.9512	**0.7228**	0.5406	0.9518
Double encoder convolutional model	Pd	**0.9682**	**1.0000**	**0.6154**	**0.9991**
	Ni	0.9957	0.9565	0.9565	0.9963
	Cu	0.9469	0.2000	**0.8571**	**0.9760**
	Ag	0.9873	0.5714	**0.5714**	**0.9889**
	Cl	**0.9299**	0.7869	0.7059	0.9557
	Br	0.9533	0.7778	0.5676	0.9744
	Average	0.9636	0.7154	**0.7123**	**0.9817**

Supplementary Note 4.2 Mechanistic studies



Supplementary Figure 24. Results of the analysis of various catalytic systems. (a) ESI-MS online monitoring of Ni-catalyzed C-S cross-coupling reaction. (b) ESI-MS online monitoring of Pd-catalyzed hydrothiolation reaction. (c) ESI-MS offline monitoring of Cu-catalyzed Chan-Lam reaction. (d) ESI-MS offline monitoring of leaching process from Pd/C heterogeneous catalyst.

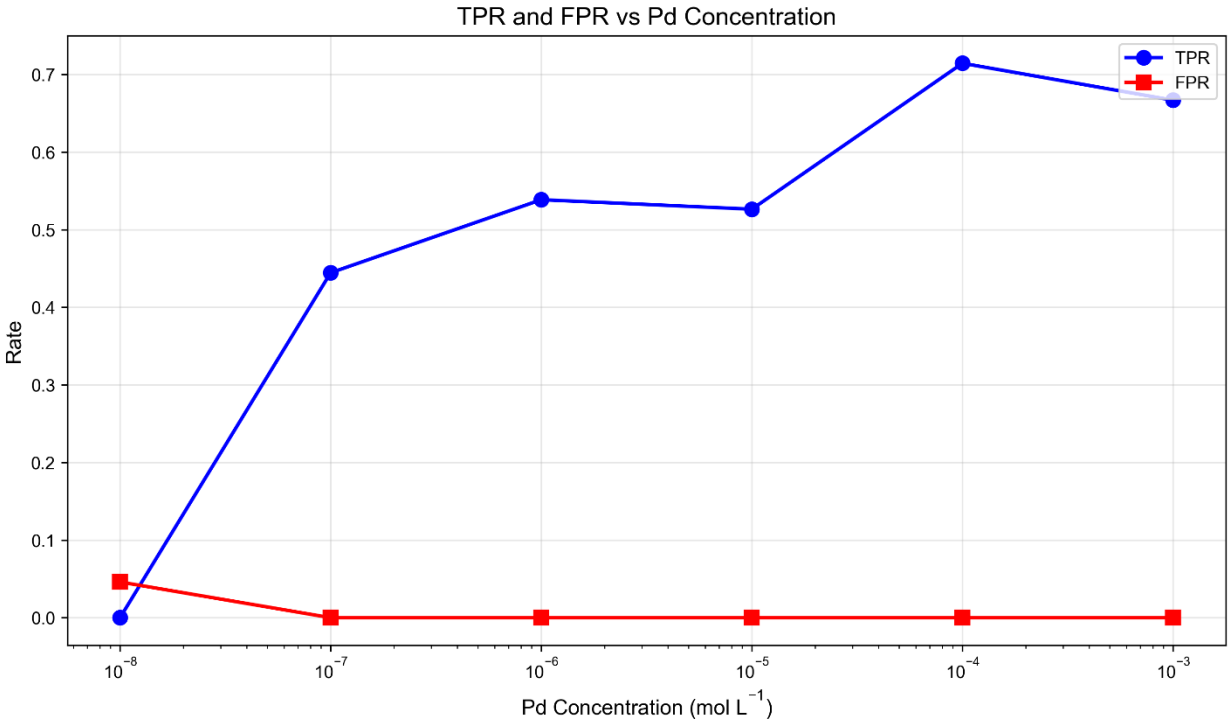
Supplementary Note 4.3 Validation in the low-concentration range

We prepared solutions of IMesPd(acac)Cl in different concentrations and obtained the corresponding mass spectra. Next, we processed them through our pipeline using the neural network trained on augmented data (see **Supplementary Notes 3.5–3.6**) and detected Pd-containing distributions. Results are presented in Figure 6 in the Manuscript.

In addition, we manually labeled all Pd-containing distributions in recorded spectra and calculated true-positive rate (TPR) and false-positive rate (FPR) for our model's predictions. Classification threshold determined in **Supplementary Note 3.6** was used in these calculations. Results are presented in **Supplementary Table 8** and **Supplementary Figure 25**.
The model produced no false-positive predictions from 10^{-3} to 10^{-7} M; false-positive predictions first occurred at 10^{-8} M.

Supplementary Table 8. Evaluation of Pd detection pipeline for IMesPd(acac)Cl in different concentrations. Here, TP, TN, FP, and FN are numbers of true-positive, true-negative, false-positive and false-negative predictions respectively.

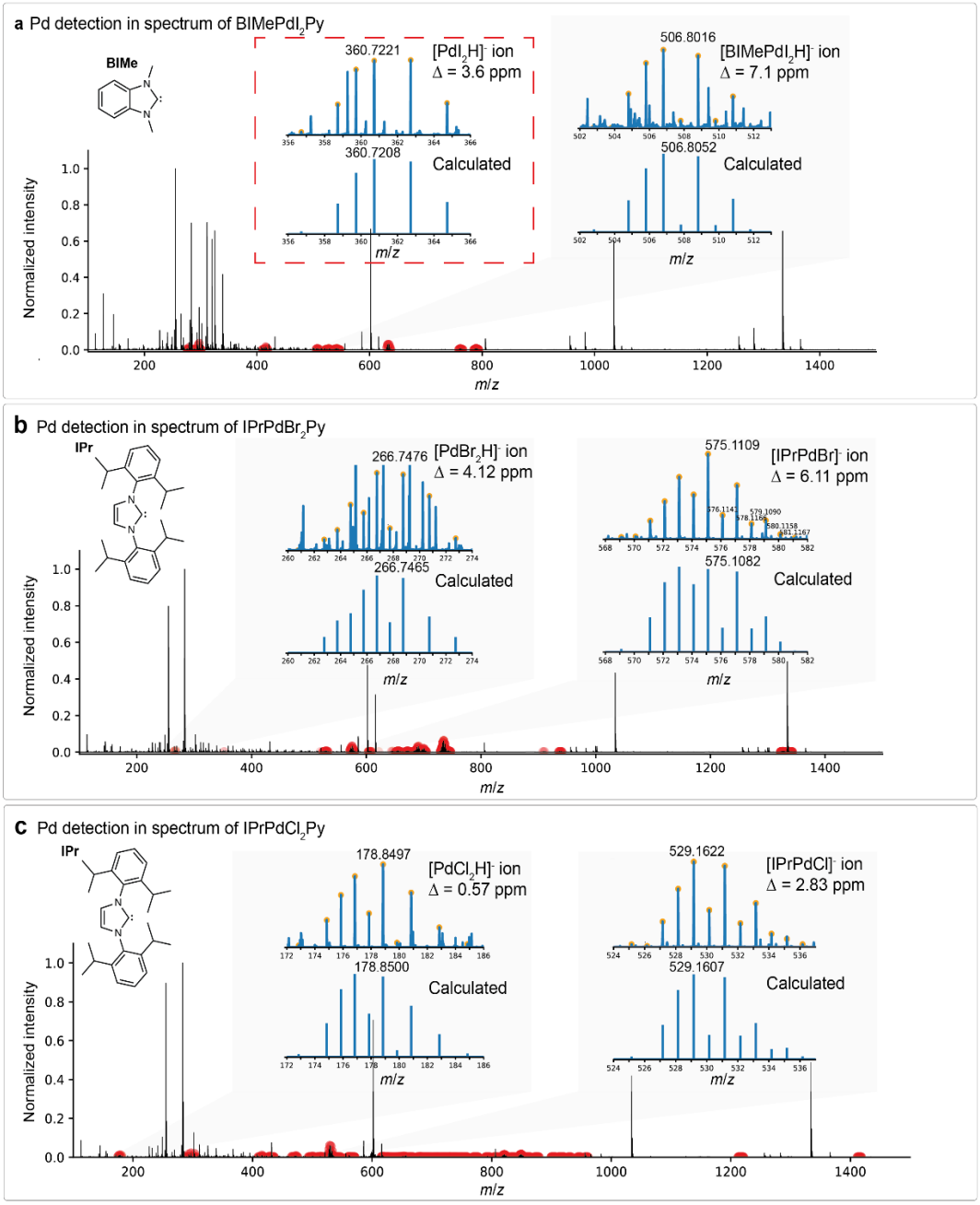
Pd concentration, (mol L^{-1})	**# of manually detected Pd distributions**	**TP**	**TN**	**FP**	**FN**	**TPR**	**FPR**	**Accuracy**
10^{-3}	27	18	16	0	9	0.6667	0	0.79
10^{-4}	21	15	17	0	6	0.7143	0	0.84
10^{-5}	19	10	22	0	9	0.5263	0	0.78
10^{-6}	13	7	56	0	6	0.5385	0	0.91
10^{-7}	9	4	77	0	5	0.4444	0	0.94
10^{-8}	2	0	83	4	2	0	0.0460	0.93
10^{-9}	0	0	103	0	0	–	0	1.00



Supplementary Figure 25. TPR and FPR values of Pd detection through Pd concentration.

Supplementary Note 4.4 Pd-H species detection

Here, the developed pipeline was used to perform detection of Pd-H species in mass spectra of 3 different Pd/NHC complexes. The algorithm successfully detects ions of interest if neighboring isotopologues differ from each other by no more than z^{-1}.



Supplementary Figure 26. MEDUSA-HR can detect Pd-H species in multicomponent mass spectra. (a) Spectrum of $BIMePdI_2Py$ in CH_3OH. (b) Spectrum of $IPrPdBr_2Py$ in CH_3OH. (c) Spectrum of $IPrPdCl_2Py$ in CH_3OH. Note, that the experimental and calculated distributions of the $[IPrPdBr]^-$ ion show a slight discrepancy due to the complexity of the spectrum.

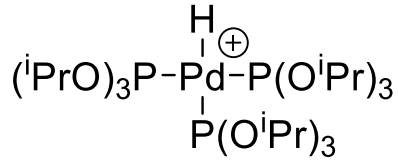
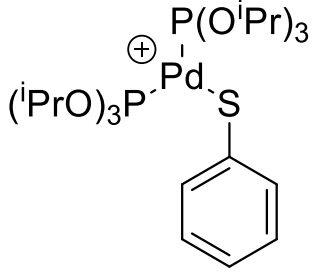
Supplementary Note 4.5 Palladium catalyzed reaction spectra analysis

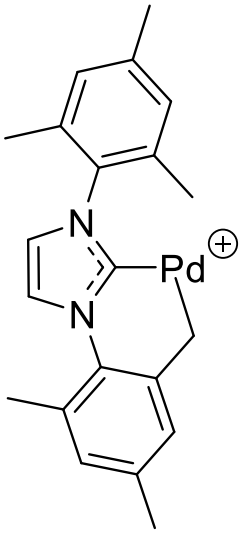
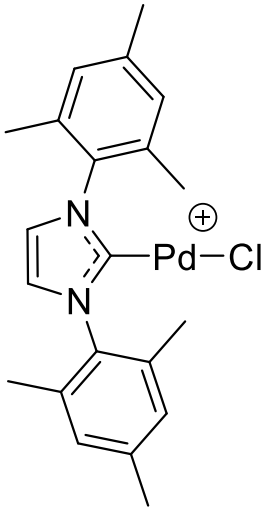
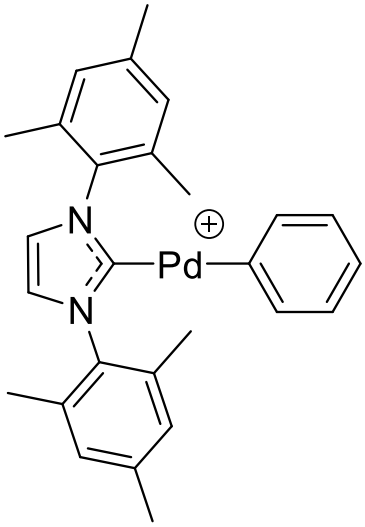
The developed approach to analysis was tested on catalytic cross-coupling reaction systems. The study included investigation of palladium complexes in 4 reaction systems:

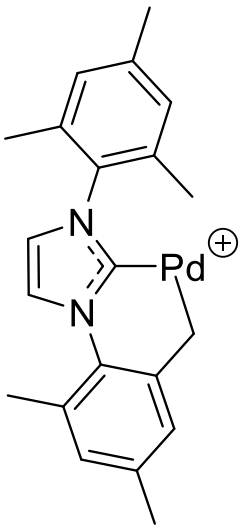
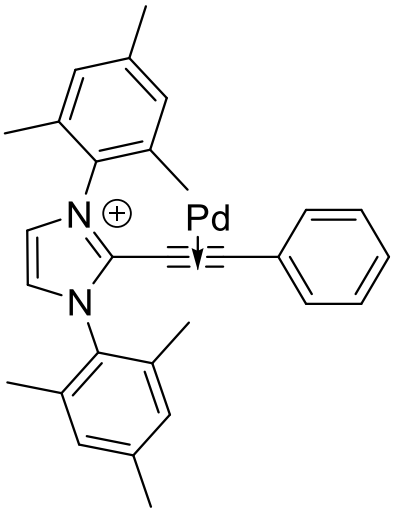
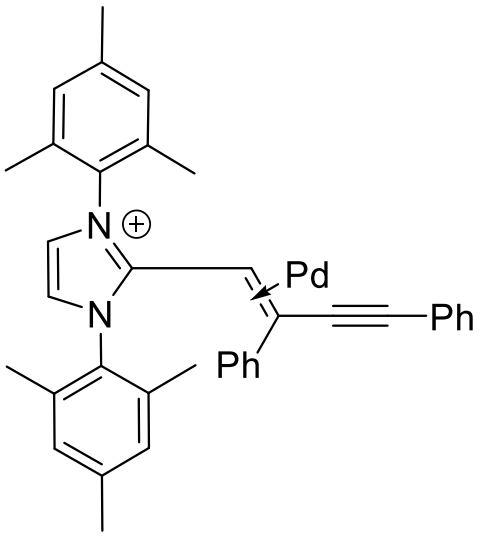
1. Disulfide addition to alkyne
2. Oxidative addition
3. Buchwald-Hartwig reaction
4. Ethynyl-NHC coupling (NHC — N-heterocyclic carbene)

The results are shown on **Supplementary Figures 27, 28**. Detected ions included **(Supplementary Table 9)**

Supplementary Table 9. Detected Pd-containing ions in catalytic reactions with automated recognition approach.

Abbreviation	**Proposed Structure**	**Reaction**
a1 (*m*/*z* = 731.2794)		Disulfide addition to alkyne
a2 (*m*/*z* = 631.1597)		Disulfide addition to alkyne

a3 (*m*/*z* = 409.0904)		Oxidative addition
a4 (*m*/*z* = 445.0668)		Oxidative addition
a5 (*m*/*z* = 487.1387)		Buchwald-Hartwig reaction

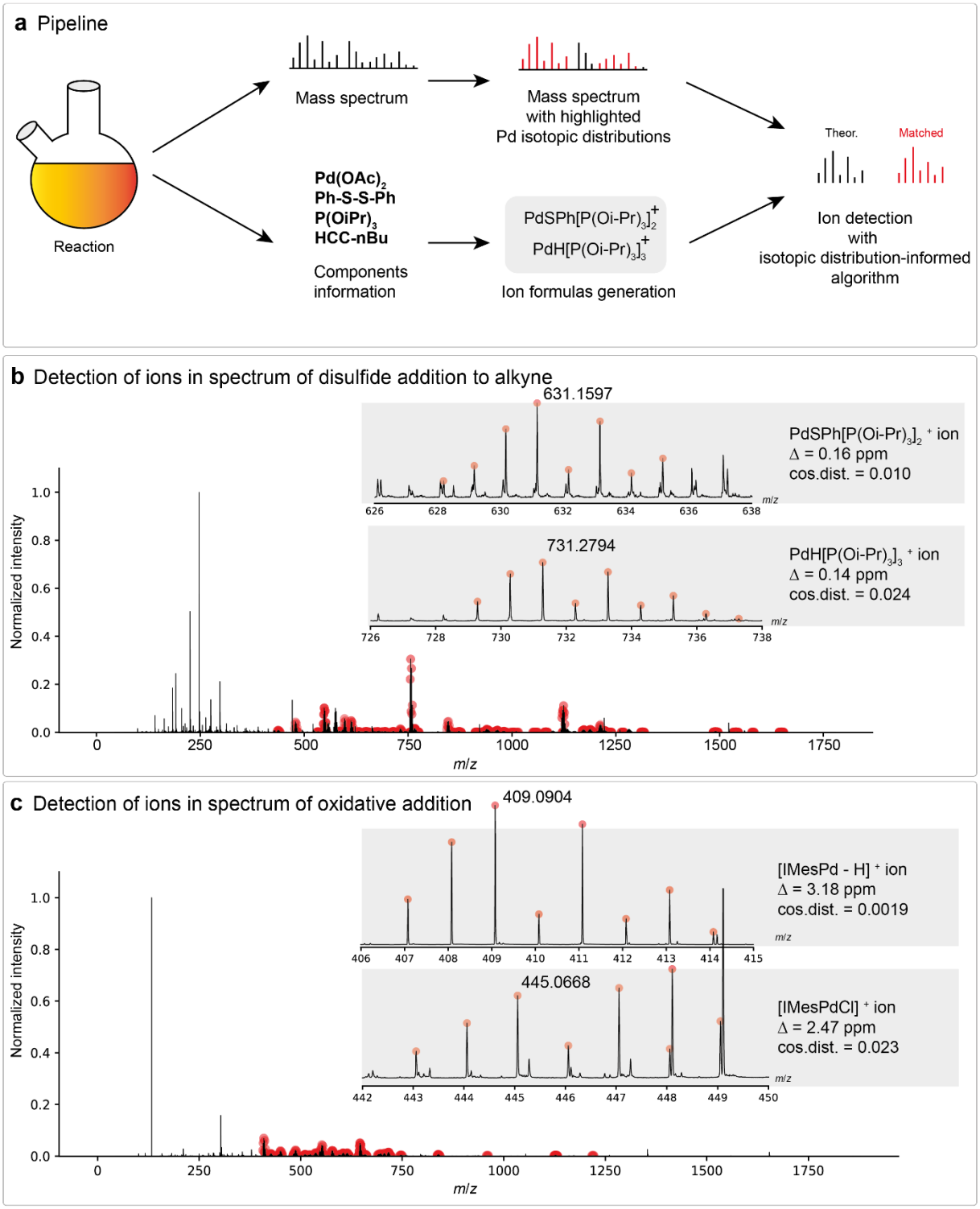
a6 (*m*/*z* = 409.0884)		Ethynyl-NHC coupling
a7 (*m*/*z* = 511.1397)		Ethynyl-NHC coupling
a8 (*m*/*z* = 613.1819)		Ethynyl-NHC coupling

In disulfide addition to alkyne reaction **a1**, **a2** ions were found. **a2** is the ionization product of a palladium complex, which is the result of the insertion of palladium through an S-S disulfide bond, with a phosphine ligand.

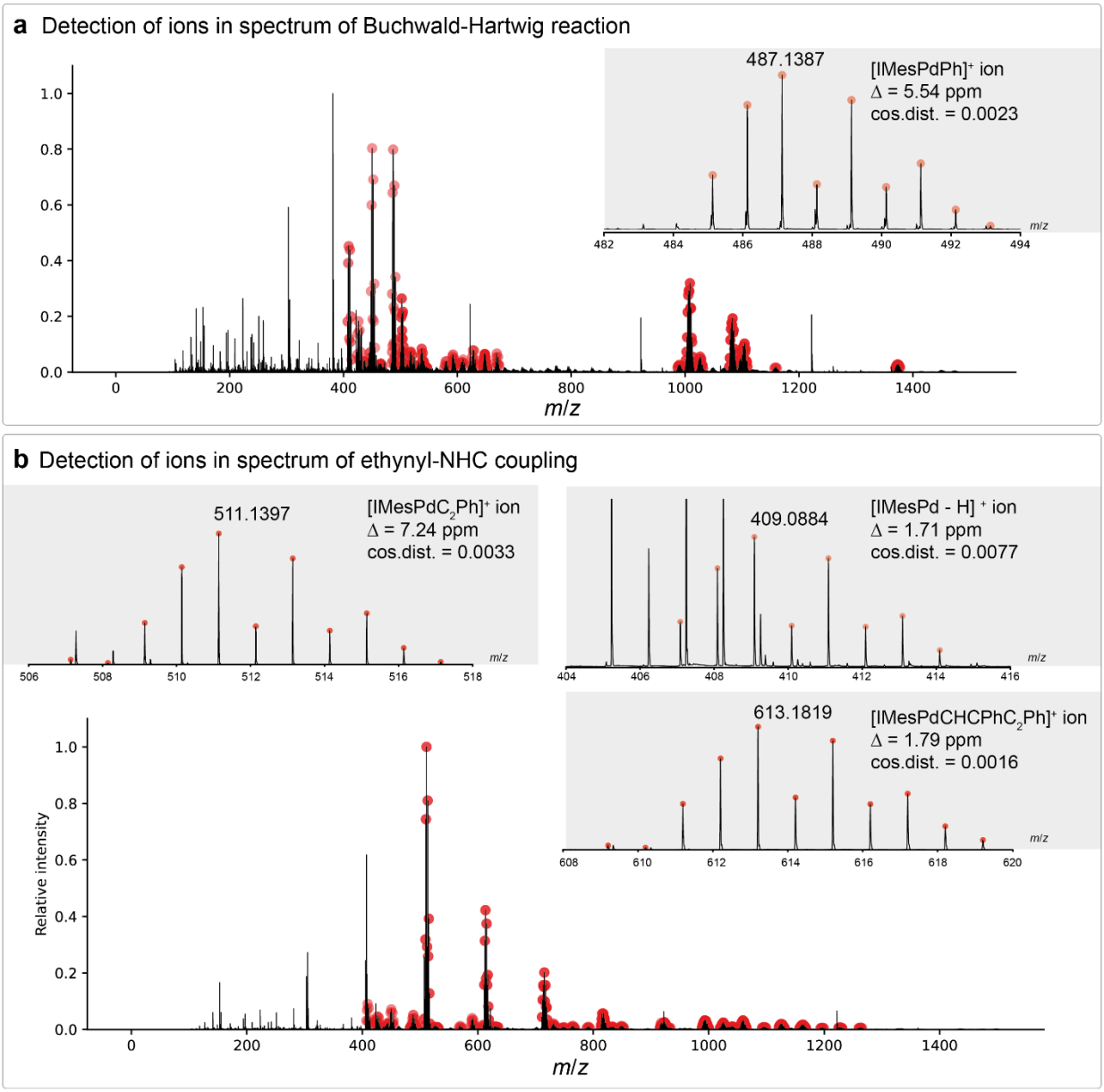
In oxidative addition reaction **a3** is a product of intramolecular palladium insertion via the C-H bond of the NHC ligand with the formation of a Pd(II) complex that lost one ligand during ionization. Ion **a4** is an ionization product of Pd(II) complex with NHC ligand. The ions **a3**, **a4** found are not specific to oxidative addition.

The formation of **a5** in the Buchwald-Hartwig reaction is the result of ionization (with bromine anion elimination) of the product of oxidative addition of bromobenzene to the NHC-Pd(0) complex.

The mechanism of ethynyl-NHC coupling was previously described in detail.[2] **a6** is a product of intramolecular palladium insertion via the C-H bond of the NHC ligand with the formation of a Pd(II) complex that lost one ligand during ionization. **a7** is either the result of palladium complex ionization prior to reductive elimination, containing an NHC ligand and an acetylene fragment, or the result of palladium coordination in the product of an ethynyl-NHC coupling[2]. Mechanism of formation of **a8** is identical to **a7** but with addition of extra phenylacetylene molecule via a triple bond of the ethynyl-NHC combination product. We also registered an ethynyl-NHC coupling product ($C_{29}H_{29}N_2^+$, *m*/*z* 405.2347).



Supplementary Figure 27. MEDUSA-HR can detect palladium containing ions in mass spectra of different reactions. (a) Pipeline for automated detection. (b) Disulfide addition reaction. (c) Oxidative addition reaction.



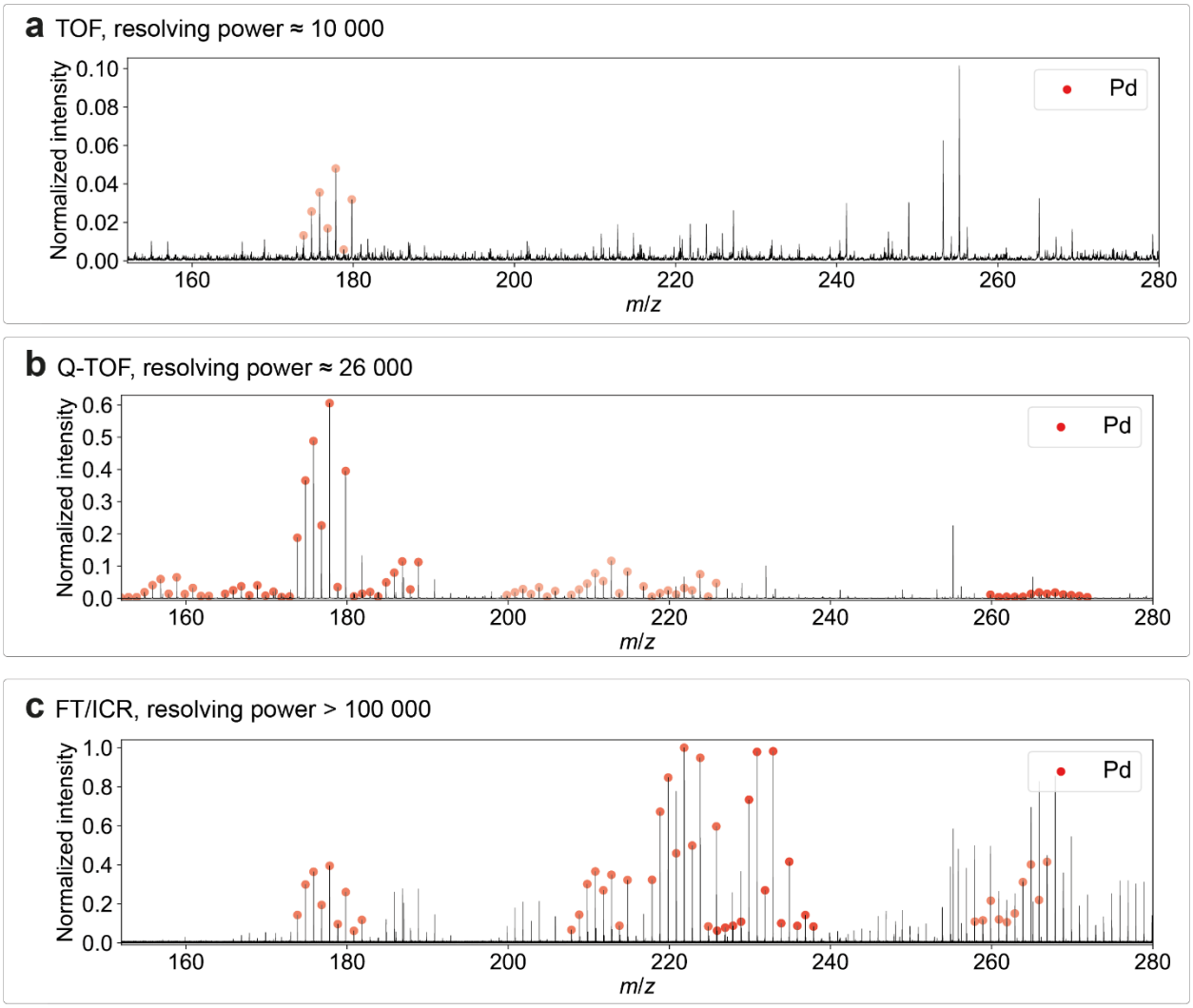
Supplementary Figure 28. MEDUSA-HR can detect palladium-containing ions in mass spectra of different reactions. (a) Buchwald-Hartwig reaction. (b) Ethynyl-NHC coupling.

Supplementary Note 4.6 Pipeline robustness for different mass analyzers.

To compare the efficiency of our full pipeline (deisotoping + elements prediction) in different experimental setups we recorded the mass spectra of $PdCl_2$ solution (concentration is 5.6×10^{-6} mol L^{-1}) using mass analyzers with different resolving powers: time-of-flight (TOF), quadrupole time-of-flight (Q-TOF) and Fourier transform ion cyclotron resonance (FT/ICR); resolving powers are ≈10 000, ≈26 000 and > 100 000 respectively.
All obtained spectra were processed through our ML pipeline and compared **(Supplementary Figure 29)**.

Key results of our comparison are:

1. Spectral analysis process starts with selecting initial signals for consideration. As different methods have different noise profiles, peak threshold becomes one of the most important hyperparameters. Specifically, after optimization we chose quantile threshold determination algorithm with 98.5, 99.5 and 99.98 percentiles for TOF, Q-TOF and FT/ICR respectively.
2. After threshold tuning, deisotoping algorithm perform well with all three mass analyzers. That demonstrates robustness of our deisotoping – classification pipeline across different mass analyzers.
3. Molecular ion was detected on all three spectra. However, additional Pd distributions were not found on the spectrum obtained from the TOF instrument. The most “balanced” spectrum was recorded with Q-TOF analyzer: additional Pd distributions are distinguishable and are not mixed with noise peaks.
4. Several Pd distributions were identified by the deisotoping algorithm with minor errors, such as missed peaks or substitution with noise peaks. Nevertheless, our double-encoder classifier, trained with augmentations, was still able to correctly detect Pd in these “corrupted” distributions. This demonstrates that our neural network is highly robust and can effectively compensate for imperfections in the deisotoping algorithm.



Supplementary Figure 29. Comparison of Pd detection pipeline performance for spectra acquired using mass analyzers with different resolving powers: Bruker microTOF (≈10,000), Bruker maXis (Q-TOF) (≈26,000), and Bruker solariX FT/ICR (>100,000).

Supplementary references

1. Gurevich, P. E. *et al*. Data supporting automatic recognition of Pd ions in high-resolution mass spectra of multicomponent samples without visible fine isotope structures. *Figshare*, https://doi.org/10.6084/m9.figshare.30455435 (2026).

2. Eremin, D. B. *et al.* Mechanistic study of Pd/NHC-catalyzed Sonogashira reaction: discovery of NHC-ethynyl coupling process. *Chem. Eur. J.* **26**, 15672–15681 (2020).