Joint structural annotation of small molecules using liquid chromatography retention order and tandem mass spectrometry data

Eric Bach[1,*], Emma L. Schymanski[2] and Juho Rousu[1,*]

[1] Department of Computer Science, Aalto University, Espoo, Finland

[2] Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, 6 avenue du Swing, L-4367 Belvaux, Luxembourg.

[*] Corresponding author: firstname.lastname@aalto.fi

Abstract

We present LC-MS^{2}Struct, a machine learning framework for structural annotation of small molecule data arising from liquid chromatography-tandem mass spectrometry (LC-MS^2) measurements. LC-MS^{2}Struct predicts the annotations for a set of mass spectrometry features in a sample, using the ions' observed retention orders and the output of state-of-the-art MS^2 scorers. LC-MS^{2}Struct is based on a novel structured prediction model trained to benefit from dependencies between retention times and the mass spectral features for an improved annotation accuracy.

We demonstrate the benefit of LC-MS^{2}Struct on a comprehensive dataset containing reference MS^2 spectra and retention times of 4327 molecules from MassBank, measured using a variety of LC conditions. We show that LC-MS^{2}Struct obtains significantly higher annotation accuracy than methods based on retention time prediction. Furthermore, LC-MS^{2}Struct improves the annotation accuracy of state-of-the-art MS^2 scorers by up to 66.1 percent and even up to 95.9 percent when predicting stereochemical variants of small molecules.

Introduction

Structural annotation of small molecules in biological samples is a challenging task and a bottleneck in various research fields including biomedicine, biotechnology, drug discovery and environmental sciences. Samples in untargeted metabolomics studies typically contain thousands of different molecules, most of which remain unidentified [1–3]. Liquid chromatography (LC) tandem mass spectrometry (LC-MS^2) is one of the most widely used analysis platforms [4], as it allows for high-throughput screening, has high sensitivity and is applicable to a wide range of molecules. Briefly, in LC-MS^2, molecules are first separated by their different physicochemical interactions between the mobile and stationary phase of the LC, resulting in retention time (RT) differences. Subsequently, separation happens according to their mass-to-charge ratio (m/z) in a mass analyzer (MS^1). Finally, the molecular ions are isolated and fragmented in the tandem mass spectrometer (MS^2), typically using a narrow mass window. For each ion, the recorded fragments and their intensities constitute what is called the MS^2 spectrum. In an untargeted LC-MS^2 workflow, large sets of MS features (MS^1, MS^2, RT), arise from a single sample. The goal in structural annotation is to associate each feature with a candidate molecular structure, for further downstream interpretation.

In recent years, many powerful methods [5, 6] to predict molecular structure annotations for MS^2 spectra have been developed [7–18]. In general, these methods find candidate molecular structures potentially associated with the MS feature, for example, by querying molecules with a certain mass from a structure database (DB) such as HMDB [19] or PubChem [20] and, subsequently, compute a matching score between each candidate and the MS^2 spectrum. The highest scoring candidate is typically considered as the structure annotation of a given MS^2. However, even the best-of-class methods only reach an annotation accuracy of around 40% [17] in evaluation when searching large candidate sets like PubChem, and therefore, in practice, a *ranked list* of molecular structures is provided to the user (e.g. top 20 structures).

Even though readily available in all LC-MS^2 pipelines and recognized as valuable information [21, 22], RT remains underutilized in automated approaches for structure annotation based on MS^2. For example, only one of the above mentioned tools provides functionality to use the RT information, namely MetFrag [11]. An explaining factor for this is that RT not only depends on the molecular structure, but also the LC conditions (e.g., mobile phase composition, column pressure, etc.) [23, 24]. Thus, a molecule generally has different RTs under different LC conditions and in different laboratories [24]. Typically, the RT information is used as post-processing for candidate lists, e.g., by comparing measured and reference standard RTs [3, 24]. This approach, however, is limited by the availability of experimentally determined RTs of reference standards. RT prediction models [25, 24], on the other hand, allow to predict RTs solely based on the candidates' molecular structure and have been successfully applied to aid structure annotation [26–29]. However, such prediction models generally have to be calibrated to the target LC configuration [3]. Calibration requires at least some amount of target LC reference RT data to be available [21, 30, 29].

Recently, the idea of predicting retention *orders* (RO), *i.e.*, the order in which two molecules elute from the LC column, has been explored [31–34]. ROs are largely preserved within a family of LC systems (*e.g.* reversed phase or HILIC). Therefore, RO predictors can be trained using a diverse set of RT reference datasets and applied to out-of-dataset LC setups with high accuracy [31]. Integration of RO and MS^2 based scores using probabilistic graphical models was shown to improve the annotation performance in LC-MS^2 experiments [34].

In this study we set out to provide a new perspective on jointly using MS^2 and RO information for the structure annotation of LC-MS^2 data. For that, we present a novel machine learning framework called LC-MS^{2}Struct, which learns to optimally combine the MS^2 and RO information for the accurate annotation of a sequence of MS features. LC-MS^{2}Struct relies on the Structured Support Vector Machine (SSVM) [35] and Max-margin Markov Network [36] frameworks. In contrast to the previous work by Bach et al. [34], our framework does not require a separately learned RO prediction model. Instead, it optimizes the SSVM parameters such that the score margin between correct and any other sequence of annotations is maximized, subject to a graphical model representing the pairwise ROs as edges and the candidate sets of molecular structures for each MS feature as candidate node labels. That means that LC-MS^{2}Struct learns to optimally use the RO information in an LC-MS^2 experiment. We trained LC-MS^{2}Struct on all available reversed phase LC data from MassBank (MB) [37], which we processed to extract ground-truth annotated (MS^2, RT)-tuples covering a diverse set of LC and MS configurations. In our experiments we evaluate LC-MS^{2}Struct across all subsets of homogeneous LC-MS^2 configurations and compare it with three other previously proposed approaches: RT filtering, log$P$ predictions [11], and RO predictions [34]. Our framework can be combined with any MS^2 scorer and applied to new LC-MS^2 data, including new LC conditions without re-training, and is demonstrated below with CFM-ID [9, 18], MetFrag [11] and SIRIUS [8, 17].

Overview of LC-MS^{2}Struct

In this section we discuss the main components of LC-MS^{2}Struct, which are also illustrated in Figure 1. Further details can be found in the Methods section.

Input and output. As input we consider a typical data setting present in an untargeted LC-MS^2 based experiments, after pre-processing such as chromatographic peak picking and alignment (Figure 1**a**). Such data comprises a sequence of MS features, here indexed by σ. Each feature consists of MS^1 information (e.g. mass, adduct and isotope pattern), LC retention time (RT) t_σ and an MS^2 spectrum x_σ. We assume that a set of candidate molecules $\mathcal{C}_\sigma$ is associated with each MS feature σ. Such a set can be, for example, generated from a structure database (e.g. PubChem [20], ChemSpider [38] or PubChemLite [39]) based on the ion's mass, a suspect list, or an in silico molecule generator (e.g. SmiLib v2.0 [40, 41]). We furthermore require that for MS^2 spectrum x_σ, a matching score $\theta(x_\sigma, m)$ with its candidates $m \in \mathcal{C}_\sigma$ is pre-computed using an in silico tool, such as CFM-ID [9, 18], MetFrag [11] or SIRIUS [8, 17]. LC-MS^{2}Struct predicts a score for MS feature σ and each associated candidate $m \in \mathcal{C}_\sigma$ based sequence of spectra $\mathbf{x} = (x_\sigma)_{\sigma=1}^L$, of length L, and the ROs derived from the observed RTs $\mathbf{t} = (t_\sigma)_{\sigma=1}^L$. These scores are used to rank the molecular candidates associated with the MS features (Figure 1**b**).

Candidate ranking using max-marginals. We define a fully connected graph $G = (V, E)$ capturing the MS features and modelling their dependencies (Figure 1**c**). Each node $\sigma \in V$ corresponds to a MS feature, and is associated with the pre-computed MS^2 matching scores $\theta(x_\sigma, m)$ between the MS^2 spectrum x_σ and all molecular candidates $m \in \mathcal{C}_\sigma$. The graph G contains an edge $(\sigma, \tau) \in E$ for each MS feature *pair*. A scoring function F is defined predicting a compatibility score between a sequence of molecular structure assignments $\mathbf{y} = (y_\sigma)_{\sigma=1}^L$ in the label-space $\Sigma = \mathcal{C}_1 \times \ldots \times \mathcal{C}_L$ and the observed data:

$$F(\mathbf{y} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G) = \underbrace{\frac{1}{|V|} \sum_{\sigma \in V} \theta(x_\sigma, y_\sigma)}_{\text{Node scores: MS}^2\text{ information}} + \underbrace{\frac{1}{|E|} \sum_{(\sigma,\tau) \in E} f\left((t_\sigma, t_\tau), (y_\sigma, y_\tau) \,|\, \mathbf{w}\right)}_{\text{Edge scores: RO information}}, \tag{1}$$

where the function f outputs an edge score for each candidate assignment pair (y_σ, y_τ) given the observed RTs (t_σ, t_τ) and the derived RO (Figure 1**d**). The edge score expresses the agreement between the observed and the predicted RO for a candidate pair, *i.e.* if a candidate pair receives a high score it is more likely to be correct. Function f is parameterized by the vector $\mathbf{w}$, which is trained specifically for each MS^2 scorer (see next section). Using the compatibility score function F (Equation (1)) we compute the max-marginals [42] for each candidate and MS features. The max-marginal score of a particular candidate $m \in \mathcal{C}_\sigma$ and MS feature σ is defined as the maximum compatibility score that a candidate assignment $\bar{\mathbf{y}} \in \Sigma$ with $\bar{y}_\sigma = m$ can reach:

$$\mu(y_\sigma = m \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G) = \max_{\{\bar{\mathbf{y}} \in \Sigma \,:\, \bar{y}_\sigma = m\}} F(\bar{\mathbf{y}} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G).$$

We use μ to rank the molecular candidates [34]. For general graphs G the max-marginal inference problem (MMAP) is intractable in practice due to the exponential size of the label space Σ. Therefore, we approximate the MMAP problem by performing the inference on tree-like graphs T_k randomly sampled from G (Figure 1**c**), for which exact inference is feasible [42, 43]. Subsequently, we average the max-marginal scores $\mu(y_\sigma = m \,|\, \mathbf{x}_i, \mathbf{t}_i, \mathbf{w}_k, T_k)$ over a set of trees $\mathbf{T}$, an approach that performed well for practical applications [44, 45, 34]. For each spanning tree T_k, we apply a separately trained SSVM model $\mathbf{w}_k$ to increase the diversity of the predictions.

Joint annotation using Structured Support Vector Machines (SSVM). We propose to tackle the joint assignment of candidate labels $\mathbf{y} \in \Sigma$ to the sequence of MS features of a LC-MS^2 experiment through structured prediction, a family of machine learning methods generally used to annotate sequences or networks [35, 46, 45]. In our model, the structure is given by the observed RO of the MS feature pairs (y_σ, y_τ), which provides additional information on the correct candidate labels y_σ and y_τ. Given a set of annotated LC-MS^2 experiments extracted from MassBank [37] (Figure 1**e**), we train a Structured Support Vector Machine (SSVM) [35] model $\mathbf{w}$ predicting the edge scores. SSVMs models can be optimized using the max-margin principle [35]. In a nutshell, given a set of ground truth annotated MS feature sequences, the model parameters $\mathbf{w}$ are optimized such that the correct label sequence $\mathbf{y}_i \in \Sigma_i$, that is the structure annotations for all MS features in an LC-MS^2 experiment, scores higher than any other possible label sequence assignment $\mathbf{y} \in \Sigma_i$ (Figure 1**f**).

Results

This section describes our experiments and the corresponding results with LC-MS^{2}Struct. We start with a description of the training and evaluation data extracted from MassBank. Then, we continue with a comparison of LC-MS^{2}Struct to other approaches for MS^2 and RT or RO score integration. Subsequently, we go into more details by analysing the performance of LC-MS^{2}Struct for different molecular classes. We conclude with a study of our method applied for the ranking of candidate sets including stereoisomers.

Extracting training data from MassBank. For this study we extracted ground truth annotated MS^2 spectra and RTs from MassBank [37], a public online database for MS^2 data. Each individual MassBank record typically provides a rich set of meta information (see Extended Data), such as the chromatographic and MS conditions as well as molecular structure annotations. To train the SSVM model of LC-MS^{2}Struct, we need sets of MS features, *i.e.* (MS^2, RT)-tuples, with ground truth structure annotations as available in MassBank. We process the MassBank data such that the experimental conditions are consistent *within* each MS feature set. That means, for example, that the LC setup is identical, such that we can compare the RTs within the set to derive the ROs, or that the same MS configuration was used, as we would assume in a typical LC-MS^2 experiment. We developed a Python package “massbank2db” [47] that can process MassBank records and groups them into consistent MS feature sets, which we denote as MB-subsets. For the SSVM training and the evaluation of LC-MS^{2}Struct, as well as comparison methods, we sample sequences of MS features to simulate LC-MS^2 experiments in which we measure the signal of multiple unknown compounds under consistent experimental setups. Figure 1**e** illustrates the grouping and LC-MS^2 sampling process. Two collections of MassBank data were considered: ALLDATA and the ONLYSTEREO subset. Further details can be found in the Methods section.

Comparison of LC-MS^{2}Struct with other approaches. In the first set of experiments we compare LC-MS^{2}Struct with previous approaches for candidate ranking either using only MS^2 or additionally RT or RO information: *Only-MS^2* uses the MS^2 spectrum information to rank the molecular candidates and serves as baseline; *MS^2+RO* [34] uses a Ranking Support Vector Machine (RankSVM) [48, 49] to predict the ROs of candidate pairs and a probabilistic inference model to combine the ROs with MS^2 scores; *MS^2+RT* uses predicted RTs to remove false positive molecule structures from the candidate set, ordered by their MS^2 score, by comparing the predicted and observed RT; *MS^2+logP* is an approach introduced by Ruttkies et al. [11], which uses the observed RT to predict the XLogP3 value [50] of the unknown compound and compares it with the candidates’ XLogP3 values extracted from PubChem to refine the initial ranking based on the MS^2 scores. A detailed description of the comparison approaches can be found in the Methods section. The RO based methods (LC-MS^{2}Struct and MS^2+RO) were trained using the RTs from all available MB-subsets, at the same time ensuring that no test molecular structure (based on InChIKey first block) was used for the model training (structure disjoint). On the other hand, for the RT based approaches (MS^2+RT and MS^2+log$P$), the RT and XLogP3 predictors were trained in a structure disjoint fashion, using only the RT data available for that respective MB-subset. For the experiment, all MB-subsets with more than 75 (MS^2, RT)-tuples from the ALLDATA data setup were used, as the RT based approaches require target LC system-specific RT training data (see Extended Data). The ranking performance was computed for each LC-MS^2 experiment within a particular MB-subset. The molecules in the candidate sets are identified by their InChIKey first block (*i.e.* the structural skeleton). That means, there are no stereoisomers in the candidate set and the rank of the ground truth molecular structure is determined using the InChIKey first block. Each candidate ranking approach was evaluated with three state-of-the-art MS^2 scorers: CFM-ID 4.0 [18], MetFrag [11] and SIRIUS [17]. Further details can be found in the Methods section.

Figure 2**a** shows the average ranking performance (top-k accuracy) across 350 LC-MS^2 experiments, with each encompassing about 50 (MS^2, RT)-tuples (see Methods). For CFM-ID and MetFrag, LC-MS^{2}Struct provides 3.1 and 4.5 percentage unit increases over the Only-MS^2 for the top-1 accuracy, corresponding to 53.5% and 66.1% performance gain. In our setting, that translates to 1.6 respectively 2.3 additional identifications at the top rank (out of approx. 50). The performance improvement increases for larger k, reaching as far as 7.2 and 8.6 percentage units at top-20, which means 3.6 respectively 4.3 additional correct structures in the top-20. For SIRIUS, the improvements are only modest, on average around 0.5 percentage units for top-1 to top-20. The runner-up score integration method is MS^2+RO, which also makes use of predicted ROs. Combined with SIRIUS, MS^2+RO actually achieves the best molecule ranking performance of all considered methods. For CFM-ID and MetFrag it leads to about half of the performance gain as LC-MS^{2}Struct. The approaches relying on RTs, either by candidate filtering (MS^2+RT) or through log$P$ prediction (MS^2+log$P$), only lead to a tiny improvement for MetFrag and CFM-ID, but none for SIRIUS, for which we even observe MS^2+RT leading to a decrease in ranking performance by about 2 percentage units. An explanation for this is that the filtering approach removes on average 4.7% of the correct candidates, which leads to false negative predictions.

The performance gain by using either RO or RT varies between the MB-subsets that differ by their LC-MS^2 setup (see Supplementary Table 4) and compound class composition (see Extended Data). We illustrate these differences in Figure 2**b**. Applying LC-MS^{2}Struct improves the ranking performance in almost all MB-subsets, including the SIRIUS data (some very slight decreases were observed in some SIRIUS sets). This is in stark contrast to the RT based approaches (MS^2+RT and MS^2+log$P$), which often lead to less accurate rankings, especially for SIRIUS. Furthermore, as can be seen already from the average results (Figure 2**a**), the benefit of LC-MS^{2}Struct depends on the MS^2 base scorer. For example, the top-1 accuracy of the subsets “AC_003” and “NA_003” can be greatly improved for MetFrag but show little or no improvement for CFM-ID. Interestingly, both datasets are natural product toxins, which are perhaps poorly explained by the bond-disconnection approach of MetFrag (often observed for substances with many rearrangements). On the other hand, for “RP_001” and “LQB_000” the largest improvements can be reached for CFM-ID. The RT filtering approach (MS^2+RT) performs particularly well for “LQB_000” and “UT_000”. These subsets are characterized by a relatively homogeneous set of molecules in terms of ClassyFire [51] super-classes (see Extended Data), encompassing mostly lipids and lipid-like molecules. Since the RT prediction models are trained using only data from the respective MB-subset, this can lead to more accurate models for subsets with less heterogeneous sets of molecules. Hence, the RT filtering could work well in such cases [26].

Performance analysis of LC-MS^{2}Struct for different compound classifications. Our next experiment investigates how LC-MS^{2}Struct can improve the identification across different categories in two molecule classification systems. The first system is the ClassyFire [51] taxonomy, which we use to assign molecule classes to all ground truth structures in our evaluation set. As a second classification system, we use the one provided by PubChemLite [39]. Figure 3 shows the average top-1 and top-20 accuracy improvement of LC-MS^{2}Struct over the Only-MS^2 baseline for each ClassyFire super-class and PubChemLite annotation category (see Methods). For ClassyFire (Figure 3**a**), we observe that the ranking performance improvement for the different super-classes depends on the MS^2 scorer. For example, the top-1 accuracy of “Alkaloids and derivatives” can be improved by 6.7 percentage units for MetFrag, but improves only very little for CFM-ID and SIRIUS (about 1 percentage unit). The picture looks different for “Organic oxygen compounds”, for which the top-1 accuracy improves by about 4.7 percentage units when using CFM-ID, but little to no improvement is observed for the other MS^2 scorers. This suggests that the CFM-ID results may be improved with the inclusion of more “Organic oxygen compounds”. On the other hand, it seems that the “Alkaloids and derivatives”, “Organic acids and derivatives” and “Organic nitrogen compounds” may be less well explained by MetFrag (perhaps with more rearrangements, or less distinguishable spectra), such that the improvement from the RO approach is more apparent.

For the PubChemLite classification (Figure 3**b**) we also see that different MS^2 scorers benefit differently by using LC-MS^{2}Struct. The improvement seems generally more consistent across the annotation categories, with one or two differing exceptions for MetFrag and CFM-ID. The SIRIUS performance seems unaffected, irrespective of the annotation category. Looking at the top-1 cases: For CFM-ID, the biggest improvement is in the “Food Related” category. For MetFrag, the category that improved the most with LC-MS^{2}Struct was “Agrochemicals”, whereas both “Agrochemicals” and “Identification” showed the least improvement for CFM-ID. The performance was relatively consistent over the other categories. For the top-20 cases, the performance seems relatively consistent except for the “Food related” (as for top-1) and “noClassification” cases. The low performance gain achieved by LC-MS^{2}Struct for molecules not covered in PubChemLite (“noClassification”) could be due to the fact that one third of the “noClassification” molecules belong to the ClassyFire class “Glycerophospholipids”. As shown in Extended Data Figure 6, this class does not benefit from LC-MS^{2}Struct, unlike other lipid classes also shown in that figure.

Annotation of stereoisomers. In general, MS^2 alone cannot reliably distinguish between stereoisomers [5, 24]. Thus MS^2 scorers mostly output the same matching score between spectrum and candidate molecule for different stereoisomers (*c.f.* [7, 17]). However, there is a difference between stereoisomers that vary in their double-bond orientation (*e.g.* *cis-trans* or *E-Z* isomerism), which may have different shapes and thus exhibit different fragmentation and/or interactions with the LC system in some cases (see Figure 5**a**), compared with stereoisomers involving chiral centres (*e.g.* *R, S* isomers), which may not exhibit such dramatic differences in regular LC-MS^2 experiments. Thus, in our last experiment we study whether LC-MS^{2}Struct can annotate stereoisomers more accurately than MS^2 alone. For that we consider candidate sets containing stereoisomers and evaluate LC-MS^{2}Struct only using MassBank records where the ground truth structure has stereochemistry information provided, *i.e.* where the InChIKey second block is not “UHFFFAOYSA” (the ONLYSTEREO data setup, see Methods). The molecular candidates are represented using two different molecular fingerprint features: One that includes stereochemistry information (3D); and one that omits it (2D) (see Methods). This allows us to assess the importance of the stereochemistry encoding of features for the candidate ranking.

Figure 4**a** shows the ranking performance of LC-MS^{2}Struct, using 2D respectively 3D fingerprints, compared with the Only-MS^2 baseline. It can be seen that LC-MS^{2}Struct improves the ranking for all three MS^2 scorers. The improvement, however, is notably larger when using candidate features that encode stereochemistry (3D). That demonstrates that LC-MS^{2}Struct can use the RO information to improve the annotation of stereoisomers, but that the molecular features need to encode stereochemistry to achieve the best performance. When looking into the top-1 performance of LC-MS^{2}Struct (3D) for the individual MS^2 scorers, we observe an improvement by 2.6, 3.8 and 3.2 percentage units for CFM-ID, MetFrag and SIRIUS, respectively. This translates to performance gains of 87.3%, 95.9% and 44.3% with about 1.5 additional structures correctly ranked at top rank (1) for all three MS^2 scorers. In contrast to our previous experiments, we see that LC-MS^{2}Struct can also improve the ranking when SIRIUS is used as MS^2 scorer.

Discussion

We have presented LC-MS^{2}Struct, a novel approach for the integration of tandem mass spectrometric and liquid chromatography data for the structural annotation of small molecules. The method learns from the pairwise dependencies in the retention order of MS features within similar LC configurations and can generalize across different, heterogeneous LC configurations. The annotation accuracies are far superior to more traditional retention time (RT) filtering and log$P$-based approaches, and also markedly better than previous methods that rely on retention orders. In particular, compared to Bach et al. [34], who used a graphical model as a post-hoc integration tool of MS^2 scores and retention order predictions, the benefits of learning the parameters of the graphical model are clear. We note that it would in principle be possible to also train the MS^2 score part (the node scores) of the model, instead of relying on separate MS^2 scorers such as SIRIUS, MetFrag and CFM-ID. Such an approach could potentially further improve the results by learning from dependencies between MS^2 and RO features. However, as the MS^2 scorers used here are already relatively mature and well-known in the community, we have left this research line open for future efforts.

Most MS^2 scorers neglect stereochemistry, or collapse their results into one result for all stereoisomers by InChIKey first block. In our experiments, we could demonstrate that LC-MS^2Struct can improve the identification of stereoisomers. The top-1 accuracy increased by 2.6 to 3.8 and the top-20 by even 4.6 to 9.2 percentage units. Furthermore, we demonstrated that the encoding of stereochemical features in the molecule representation is essential to improved the identification of stereoisomers. These can be split into two general cases: those features encoding double-bond stereochemistry (SMILES: "\" and "/") as well as the chiral centre configuration (SMILES: "@" and "@@"). Inspecting individual examples revealed that LC-MS^2Struct can separate the former cases with varying double-bond stereochemistry - *i.e.* *E/Z*- and *cis/trans*-isomers (see *e.g.* Figure 5). However, we note that there were very few examples of double-bond and/or chiral isomers measured on the same LC system in our dataset, which makes it difficult to verify these initial results, or interrogate these further - until such data is publicly available. Certain stereoisomers differing only in chiral centres (*i.e.* containing "@" and "@@") can generally only be separated using chiral column chromatography. MassBank, and hence our datasets, currently does not cover such columns. Since MassBank also contains many metabolomics (biological) datasets with primarily naturally-observed chiral forms, some of the observed improvement could also be related to biases in our dataset. In other words, certain chiral configurations might be over-represented in public databases (*i.e.* in this case MassBank), hence these are more likely to be predicted. Overall, these results suggest that LC-MS^2 annotation may be improved by the use of stereochemistry information, but that a selective fingerprint definition capturing only the stereochemistry that is relevant for non-chiral LC systems should be used or developed to investigate this further.

We developed a processing pipeline to extract ground truth annotated MS^2 spectra with RT information from MassBank. The (MS^2, RT)-tuples are grouped into subsets with homogeneous MS- and LC-conditions. This enables researchers to use MassBank data in a format suitable for machine learning, and hence can facilitate the develop of novel approaches integrating MS^2 and RT information for structure annotation. We made the pipeline available to the research community in a separate Python package "massbank2db" [47].

Methods

Notation. We use the following notation to describe LC-MS^2Struct:

$$\begin{array}{lll}
\text{Sequence of spectra} & \mathbf{x} = (x_1, \ldots, x_L) & \text{with } x_\sigma \in \mathcal{X} \\
\text{Sequence of retention times} & \mathbf{t} = (t_1, \ldots, t_L) & \text{with } t_\sigma \in \mathbb{R}_{\geq 0} \\
\text{Sequence of candidate sets} & \boldsymbol{\mathcal{C}} = (\mathcal{C}_1, \ldots, \mathcal{C}_L) & \text{with } \mathcal{C}_\sigma \subseteq \mathcal{Y} \\
\text{Sequence of labels} & \mathbf{y} = (y_1, \ldots, y_L) \in \Sigma & \text{with } y_\sigma \in \mathcal{Y} \\
\text{Candidate assignment space} & \Sigma = \mathcal{C}_1 \times \ldots \times \mathcal{C}_L, &
\end{array} \quad (312)$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote the MS^2 spectra and the molecular structure space, respectively, and $\mathcal{C}$ denotes a candidate set that is a sub-set of all possible molecular structures, and $A \times B$ denotes cross product of two sets A and B. For the purpose of model training and evaluation, we assume a dataset with ground truth labeled MS feature sequences: $\mathcal{D} = \{((\mathbf{x}_i, \mathbf{t}_i), \boldsymbol{\mathcal{C}}_i, \mathbf{y}_i)\}_{i=1}^N$, where N denotes the total number of sequences. We use $i, j \in \mathbb{N}_{\geq 0}$ to index MS feature sequences and $\sigma, \tau \in \mathbb{N}_{\geq 0}$ as indices for individual MS features within a sequence, e.g. $x_{i\sigma}$ denotes the MS^2 spectrum at index σ in the sequence i. The length of a sequence of MS features is denoted with L. We denote the ground truth labels (candidate assignment) of sequence i with $\mathbf{y}_i$ and any labelling with $\mathbf{y}$. Both, $\mathbf{y}_i$ and $\mathbf{y}$ are in Σ_i. We use y to denote the candidate label variable, whereas m denotes a particular molecular structure. For example, $y_\sigma = m$ means, that we assign the molecular structure m as label to the MS feature σ.

Graphical model for joint annotation of MS features. We consider the molecular annotation problem for the output of an LC-MS^2, that means assigning a molecular structure to each MS feature, as a structured prediction problem [35, 46, 45], relying on a graphical model representation of the sets of MS features arising from an LC-MS^2 experiment. For each MS feature σ we want to predict a label y_σ from a fixed and finite candidate (label) set $\mathcal{C}_\sigma$. We model the observed retention orders (RO) between each MS feature pair (σ, τ) within an LC-MS^2 experiment, as pairwise dependencies of the features. We define an undirected graph $G = (V, E)$ with the vertex set V containing a node σ for each MS feature and the edge set E containing an edge for each MS feature pair $E = \{(\sigma, \tau) \,|\, \sigma, \tau \in V, \sigma \neq \tau\}$ (c.f. Figure 1**a** and **c**). The resulting graph is complete with an edge between all pairs of nodes. This allows us to make use of arbitrary pairwise dependencies, instead of limiting to, say, adjacent retention times. This modeling choice was previously shown to be beneficial by Bach et al. [34]. Here we extend that approach by learning from the pairwise dependencies to optimize joint annotation accuracy, which leads to markedly improved annotation accuracy.

For learning, we define a scoring function F that, given the input MS feature sequences $(\mathbf{x}, \mathbf{t})$ and its corresponding sequence of candidate sets $\boldsymbol{\mathcal{C}}$, computes a compatibility score between the measured data and *any* possible sequence of labels $\mathbf{y} \in \Sigma$:

$$F(\mathbf{y} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G) = \frac{1}{|V|} \sum_{\sigma \in V} \theta(x_\sigma, y_\sigma) + \frac{1}{|E|} \sum_{(\sigma,\tau) \in E} \langle \mathbf{w}, \Gamma(\mathbf{t}^{\sigma\tau}, \mathbf{y}^{\sigma\tau}) \rangle, \tag{2}$$

where $\theta : \mathcal{X} \times \mathcal{Y} \rightarrow (0, 1]$ is a function returning an MS^2 matching score between the spectrum x_σ and a candidate $y_\sigma \in \mathcal{C}_\sigma$, $\langle \cdot, \cdot \rangle$ denotes the inner product, and $\mathbf{w}$ is a model weight vector to predict the RO matching score, based on the joint feature vector $\Gamma : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathcal{Y} \times \mathcal{Y} \rightarrow \mathcal{F}$ between the observed RO derived from $\mathbf{t}^{\sigma\tau} = (t_\sigma, t_\tau)$ and a pair of molecular candidates $\mathbf{y}^{\sigma\tau} = (y_\sigma, y_\tau)$.

Equation (2) consists of two parts: (1) A score computed over the nodes in G capturing the MS^2 information; and (2) a score expressing the agreement of observed and predicted RO computed over the edge set. We assume that the node scores are pre-computed by a MS^2 scorer such as CFM-ID [18], MetFrag [11] or SIRIUS [17]. The node scores are normalized to $(0, 1]$ within each candidate set $\mathcal{C}_\sigma$. The edge scores are predicted for each edge (σ, τ) using the model $\mathbf{w}$ and the joint-feature vector Γ:

$$\begin{aligned} f(\mathbf{t}^{\sigma\tau}, \mathbf{y}^{\sigma\tau} \,|\, \mathbf{w}) &= \langle \mathbf{w}, \Gamma(\mathbf{t}^{\sigma\tau}, \mathbf{y}^{\sigma\tau}) \rangle \\ &= \langle \mathbf{w}, \operatorname{sign}(t_\sigma - t_\tau) \, (\phi(y_\sigma) - \phi(y_\tau)) \rangle \\ &= \operatorname{sign}(t_\sigma - t_\tau) \langle \mathbf{w}, \phi(y_\sigma) - \phi(y_\tau) \rangle, \end{aligned} \tag{3}$$

with $\phi : \mathcal{Y} \rightarrow \mathcal{F}_\mathcal{Y}$ being a function embedding a molecular structure into a feature space. The edge prediction function (3) will produce a height edge score, if the observed RO (*i.e.* $\operatorname{sign}(t_\sigma - t_\tau)$) agrees with the predicted one.

Using the compatibility score function (2) the predicted joint annotation for $(\mathbf{x}, \mathbf{t})$ corresponds to the the highest scoring label sequence $\hat{\mathbf{y}} \in \Sigma$: $\hat{\mathbf{y}} = \arg\max_{\bar{\mathbf{y}} \in \Sigma} F(\bar{\mathbf{y}} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G)$. In practice, however, instead of only predicting the best label sequence, it can be useful to *rank* the molecular candidates $m \in \mathcal{C}_\sigma$ for each MS feature σ. That is because for state-of-the-art MS^2 scorers, the annotation accuracy in the top-20 candidate list is typically much higher than for the highest ranked candidate (top-1). Our framework provides candidate rankings by solving the following problem for each MS feature σ and $m \in \mathcal{C}_\sigma$:

$$\mu(y_\sigma = m \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G) = \max_{\{\bar{\mathbf{y}} \in \Sigma \,:\, \bar{y}_\sigma = m\}} F(\bar{\mathbf{y}} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, G). \tag{4}$$

Problem (4) returns a max-marginal μ score for each candidate m. That is, the maximum compatibility score *any* label sequence $\bar{\mathbf{y}} \in \Sigma$ with $\bar{y}_\sigma = m$ can achieve. One can interpret Equation (2) as the log-space representation of a unnormalized Markov Random Field probability distribution over $\mathbf{y}$ associated with an undirected graphical model G [43].

Feasible inference using random spanning trees (RST). For general graphs G the maximum a posterior (MAP) inference problem, that is finding the highest scoring label sequence $\mathbf{y}$ given an MS feature sequence, is an $\mathcal{NP}$-hard problem [52, 53]. The max-marginals inference

(MMAP), needed for the candidate ranking, is an even harder problem which is $\mathcal{NP}^{\mathrm{PP}}$ complete [53]. However, efficient inference approaches have been developed. In particular, if G is tree-like, we can efficiently compute the max-marginals using dynamic programming and the max-product algorithm [42, 43]. Such tree-based approximations have shown to be successful in various practical applications [44, 45, 34].

Here, we follow the work by Bach et al. [34] and sample a set of random spanning trees (RST) $\mathbf{T} = \{T_k\}_{k=1}^{K}$ from G, whereby K denotes the size of the RST sample. Each tree T_k has the same node set V as G, but and an edge set $E(T) \subseteq E$, with $|E(T)| = L - 1$, ensuring that T is a single connected component and cycle free. We follow the sampling procedure used by Bach et al. [34]. Given the RST set $\mathbf{T}$ we compute the averaged max-marginals to rank the molecular candidates [34]:

$$\bar{\mu}(y_\sigma = m \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, \mathbf{T}) = \frac{1}{K} \sum_{k=1}^{K} \left(\mu(y_\sigma = m \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, T_k) - \max_{\bar{\mathbf{y}} \in \Sigma} F(\bar{\mathbf{y}} \,|\, \mathbf{x}, \mathbf{t}, \mathbf{w}, T_k) \right), \tag{5}$$

where we subtract the maximum compatibility score from the marginal values corresponding to the individual trees to normalize the marginals before averaging [34]. This normalization value can be efficiently computed given the max-marginals μ. In our experiments, we train K individual models ($\mathbf{w}_k$) and associate them with the trees T_k to increase the diversity.

The Structured Support Vector Machine (SSVM) model. To train the model parameters $\mathbf{w}$ (see equation (2)), we implemented a variant of the Structured Support Vector Machine (SSVM) [36, 35]. Its primal optimization problem is given as [54]:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \\ \text{st.} \quad & F(\mathbf{y}_i \,|\, \mathbf{x}_i, \mathbf{t}_i, \mathbf{w}, G_i) - F(\mathbf{y} \,|\, \mathbf{x}_i, \mathbf{t}_i, \mathbf{w}, G_i) \geq \ell(\mathbf{y}_i, \mathbf{y}) - \xi_i \\ & \forall i \in \{1, \ldots, N\}, \forall \mathbf{y} \in \Sigma_i, \end{aligned} \tag{6}$$

where $C > 0$ being the regularization parameter, $\xi_i \geq 0$ is the slack variable for example i and $\ell : \Sigma_i \times \Sigma_i \to \mathbb{R}_{\geq 0}$ being a function capturing the loss between two label sequences. The constraint set definition (st.) of problem (6) leads to a parameter vector $\mathbf{w}$ that is trained according to the max-margin principle [36, 35, 46], that is the score $F(\mathbf{y}_i)$ of the correct label should be greater than the score $F(\mathbf{y})$ of any other label sequence by at least the specified margin $\ell(\mathbf{y}_i, \mathbf{y})$. Note that in the SSVM problem (6) a different graph $G_i = (V_i, E_i)$ can be associated to each training example i, allowing, for example, to process sequences of different length.

We solve (6) in its dual formulation and use the Frank-Wolfe algorithm [55] following the recent work by Lacoste-Julien et al. [54]. In the supplementary material we derive the dual problem and demonstrate how to solve it efficiently using the Frank-Wolfe algorithm and RST approximations for G_i. Optimizing the dual problem enables us to use non-linear kernel functions $\lambda : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ measuring the similarity between the molecular structures associated with the label sequences.

The label loss function ℓ is defined as follows:

$$\ell(\mathbf{y}_i, \mathbf{y}) = \frac{1}{|V_i|} \sum_{\sigma=1}^{L} (1 - \lambda(y_{i\sigma}, y_\sigma)).$$

and satisfies $\ell(\mathbf{y}, \mathbf{y}) = 0$ (a required property [54]), if λ is a normalized kernel, which holds true in our experiments (we used the MinMax kernel [56]).

Pre-processing pipeline for raw MassBank records. Extended Data Figure 8 illustrates our MassBank (MB) pre-processing pipeline implemented in the Python package "massbank2db" [47]. First, the MassBank records' text files were parsed and the MS^2 spectrum, ground truth annotation, RT and meta-information extracted. Records with missing MS^2, RT or annotation were discarded. We use the MB 2020.11 release for our experiments. Subsequently, we grouped the MassBank records into subsets (denoted as MB-subsets) where the (MS^2, RT)-tuples have been measured under the same LC- and MS-conditions. Extended Data Table 3 summarizes the grouping criteria. In the next step, we used the InChIKey [57] identifier in MassBank to retrieve the SMILES [58] representation from PubChem [20] (1st of February 2021), rather than using the contributor-supplied SMILES. This ensures that we use a single SMILES source for the molecular candidates and ground truth annotations. Before inserting the records into our final database, we performed three more filtering steps: (1) we removed records for which the ground truth exact mass deviated too much from the calculated exact mass based on the precursor mass-per-charge (m/z) and adduct type (larger than 20ppm); (2) we removed subsets that contain less then 50 unique molecular structures; (3) we removed all records associated with the MassBank prefix `LU` that were potential isobars (see pull-request #152 in the MassBank GitHub repository, `https://github.com/MassBank/MassBank-data/pull/152`). Supplementary Table 4 summarizes the meta-information for all generated MB-subsets.

Generating the molecular candidate sets. We used SIRIUS [8, 17] to generate the molecular candidate sets. For each MassBank record the ground truth molecular formula was used by SIRIUS to collect the candidate structures from PubChem [20]. The candidate sets generated by SIRIUS contain a single stereoisomer per candidate, identified by their InChIKey first block (structural skeleton). To study the ability of LC-MS^2Struct to annotate the stereochemical variant of the molecules, we enriched the SIRIUS candidates sets with stereoisomers. For that, the InChIKey first block of each candidate was used to search PubChem (1st of Feburary 2021) for stereoisomers. The additional molecules were then added to the candidate sets.

Pre-computing the MS^2 matching scores. For each MB-subset, MS^2 spectra with identical adduct type (e.g. `[M+H]+`) and ground truth molecular structure were aggregated. Depending on the MS^2 scorer we either merged the MS^2 into a single spectrum (CFM-ID and MetFrag) following the strategy by Ruttkies et al. [11] or we provided the MS^2 spectra separately (SIRIUS). To compute the CFM-ID (v4.0.7) MS^2 matching score we first predicted the in silico MS^2 spectra for all molecular candidate structures based on their isomeric SMILES representation using the pre-trained CFM-ID models (Metlin 2019 MSML) by Wang et al. [18]. We merged the three in silico spectra predicted by CFM-ID for different collision energies and compared them with the merged MassBank spectrum using the modified cosine similarity [59] implemented in the matchms [60] (v0.9.2) Python library. For MetFrag (v2.4.5) the MS^2 matching scores were calculated using the `FragmenterScore` feature based on the isomeric SMILES representation of the candidates. For SIRIUS, the required fragmentation trees are computed using the ground truth molecular formula of each MassBank spectrum. SIRIUS uses canonical SMILES and hence does not encode stereochemical information (canonical SMILES). Therefore, we used the same SIRIUS MS^2 matching score for all stereoisomers sharing the same InChIKey first block. For all three MS^2 scorers we normalized the MS^2 matching scores to the range [0, 1] separately for each candidate set. For the machine learning based scorers (CFM-ID and SIRIUS) we predicted the matching scores such that the associated MassBank record's ground truth structures was not used for the MS^2 scorer model training. If a MS^2 scorer failed on a MassBank record, we assigned a constant MS^2 score to each candidate.

Molecular feature representations. For LC-MS^2Struct, we used extended connectivity fingerprints with function-classes (FCFP) [61] to represent molecular structures in our experiments. We employed RDKit (v2021.03.1) for the FCFP fingerprint generation. The fingerprints were computed based on the isomeric SMILES. RDKit parameter "useChirality" was used to generate fingerprints that either encode stereochemistry (3D) or not (2D). We used *counting* FCFP fingerprints. To define the set of substructures in the fingerprint vector, we first generated all possible substructures, using a FCFP radius of two, based on a set of 50000 randomly sampled molecular candidates associated with our training data, and all the ground truth training structures, resulting in 6925 (3D) and 6236 (2D) substructures. We used 2D FCFP fingerprints in our experiments, except for the experiments focusing on the identification of stereoisomers, where we used 3D fingerprints. We used the MinMax-kernel [56] to compute the similarity between the molecules.

Computing molecular categories. For the analysis of the ranking performance for different molecular categories, we used two classification systems, ClassyFire [51], which classifies molecules according to their structure and PubChemLite [39], which focuses on molecules' relevance to exposomics. For ClassyFire, we used the "classyfireR" R package to retrieve the classification for each ground truth molecular structure in our dataset. For PubChemLite classifications, we first check for each molecular structure whether it is contained in PubChemLite by matching the InChIKey first block. We considered all 10 of the provided PubChemLite classes. If a molecular structure was not found in PubChemLite we assign it to the category "noClassification".

Training and evaluation data setups. We only considered MassBank data that has been analyzed using a LC reversed phase (RP) column. We removed molecules from the data if their measured retention time (RT) was less than three times the estimated column dead-time [62], as we considered such molecules to be non-retaining.

We considered two separate data setups. The first one, denoted by ALLDATA, used all available MassBank data to train and evaluate LC-MS^{2}Struct. This setup was used to compare the different candidate ranking approaches as well as to investigate the performance across various molecular classes. The second setup, denoted by ONLYSTEREO, used MassBank records where the ground truth molecular structure contains stereochemical information, *i.e.* where the InChIKey second block is not "UHFFFAOYSA". This setup was used in the experiments regarding the ability of LC-MS^{2}Struct to distinguish stereochemistry. In the training, we additionally used MassBank records that appear only without stereochemical information in our candidate sets, identified by the InChIKey second block equal to "UHFFFAOYSA" in PubChem. The number of available training and evaluation (MS^2, RT)-tuples per MB-subset are summarized in Extended Data Table 1.

For each MB-subset we sampled a set of LC-MS^2 experiments, i.e. (MS^2, RT)-tuple sequences, from the available evaluation data. The number of LC-MS^2 experiments (n below) depended on the number of available (MS^2, RT)-tuples (see Extended Data Table 1) as follows

$$n = \begin{cases} 0 & \text{if } |\mathcal{D}| < 30 \\ 1 & \text{if } |\mathcal{D}| \leq 75 \\ 15 & \text{if } |\mathcal{D}| \leq 250 \\ \left\lfloor \frac{|\mathcal{D}|}{50} \right\rfloor & \text{else.} \end{cases}$$

where $\mathcal{D}$ is a set of (MS^2, RT)-tuples with ground truth annotation and molecular candidate sets associated with a MB-subset. If there are less than 30 (MS^2, RT)-tuples available, we do not generate an evaluation LC-MS^2 experiment from the corresponding MB-subset. Based on this sampling scheme, we obtained 354 and 94 LC-MS^2 experiments for ALLDATA and ONLYSTEREO, respectively, for our evaluation (see Extended Data Table 1).

We trained eight ($K = 8$) separate SSVM models $\mathbf{w}_k$ for each evaluation LC-MS^2 experiment. For each SSVM model we first generated a set containing the (MS^2, RT)-tuples from all MB-subsets. Then, we removed all tuples whose ground truth molecular structure, determined by the InChIKey first block, was in the respective evaluation LC-MS^2 experiment. Lastly, we randomly sampled LC-MS^2 experiments from the training tuples, within their respective MB-subset, with a length randomly chosen from $\{4, \ldots, 32\}$ (see also Figure 1**e**) and an RST T_{ik} assigned for each MS feature sequence i. In total 768 LC-MS^2 training experiments were generated for each SSVM model. To speed up the model training, we restricted the candidate set size $|\mathcal{C}_{i\sigma}|$ of each training MS feature σ to maximum 75 candidate structures by random sub-sampling. Each SSVM model $\mathbf{w}_k$ was applied to the evaluation LC-MS^2 experiment, associated with different RSTs T_k, and the averaged max-marginal scores where used for the final candidate ranking (see Equation (5) and Figure 1**c**).

SSVM hyper-parameter optimization. The SSVM regularization parameter C was optimized for each training set separately using grid search and evaluation on a random validation set sampled from the training data's (MS^2, RT)-tuples (33%). A set of LC-MS^2 experiments was generated from the validation set and used to determine the Normalized Discounted Cumulative Gain (NDCG) [63] for each C value. The regularization parameter with the highest NDCG value was chosen to train the final model. We used the scikit-learn [64] (v0.24.1) Python package to compute the NDCG value, taking into account ranks up until 10 (NDCG@10) and defined the relevance for each candidate to be 1 if it is the correct one and 0 otherwise. To reduce the training time, we searched the optimal C^* only for SSVM model $k = 0$ and used C^* for the other models with $k > 0$.

Ranking performance evaluation. We computed the ranking performance (top-k accuracy) for a given LC-MS^2 experiment using the tie-breaking strategy described in [8]: If a ranking method assigns an identical score to a set of n molecular candidates, then all accuracies at the ordinal ranks k at which one of these candidates is found are increased by $\frac{1}{n}$. We computed a candidate score (*i.e.* Only-MS^2, LC-MS^2Struct, etc.) for each molecular structure in the candidate set. In the experiments using the ALLDATA setup we collapsed the candidates by InChIKey first block, assigning the maximum candidate score for each InChIKey first block group. The top-k accuracy was computed based on the collapsed candidate sets. In the ONLYSTEREO setup, we did not collapse the candidate sets before the top-k accuracy computation.

For the performance analysis of individual molecule categories, either ClassyFire [51] or PubChemLite [39] classes, we first computed the rank of the correct molecular structure for each (MS^2, RT)-tuple of each LC-MS^2 evaluation experiment based on Only-MS^2 and LC-MS^2Struct scores. Subsequently, we computed the top-k accuracy for each molecule category, associated with at least 50 unique ground truth molecular structures (based on InChIKey first block). As a ground truth structure can appear multiple times in our dataset, we generate 50 random samples, each containing only one example per unique structure, and computed the averaged top-k accuracy.

Comparison of LC-MS^2Struct with other approaches. We compared LC-MS^2Struct with three different approaches to integrate tandem mass spectrum (MS^2) and retention time (RT) information, namely RT filtering, log$P$ prediction and retention order prediction.

For RT filtering (MS^2+RT), we followed Aicheler et al. [26] who used the relative error $\epsilon = \frac{|\hat{t}-t_\sigma|}{t_\sigma}$, between the predicted ($\hat{t}$) and observed (t_σ) retention time. We set the filtering threshold to the 95%-quantile of the relative RT prediction errors estimated from the RT model's training data, following [27, 29]. We used scikit-learn's [64] (v0.24.1) implementation of the Support Vector Regression (SVR) [65] with radial basis function (RBF) kernel for the RT prediction. For SVR, we use the same 196 features, computed using RDKit (v2021.03.1), as Bouwmeester et al. [25].

For log$P$ prediction (MS^2+log$P$) we followed Ruttkies et al. [11] who assigned a weighted sum of an MS^2 and log$P$ score $s = \beta \cdot s_{\mathrm{MS}^2}(m) + (1-\beta)s_{\log P}(m)$ to each candidate $m \in \mathcal{C}_\sigma$, and use it rank the set of molecular candidates. The log$P$ score is given by $s_{\log P}(m) = \frac{1}{\delta\sqrt{2\pi}}\exp\left(-\frac{(\log P_m - \log P_\sigma)^2}{2\delta^2}\right)$, where $\log P_m$ is the predicted XLogP3 [50] extracted from PubChem [20] for candidate m, and $\log P_\sigma = a \cdot t_\sigma + b$ is the XLogP3 value of the unknown compound, associated with MS feature σ, predicted based on its measured RT t_σ. The parameters a and b of the linear regression model were determined using a set of RT and XLogP3 tuples associated with the LC system. As Ruttkies et al. [11], we set the $\delta = 1.5$ and set β such that it optimizes the top-1 candidate ranking accuracy, calculated from a set of 25 randomly generated training LC-MS^2 experiments.

For retention order prediction (MS^2+RO) we used the approach by Bach et al. [34] which relies on a Ranking Support Vector Machine (RankSVM) implementation in the Python library ROSVM

[31, 66] (v0.4.0). We used counting `substructure` fingerprints calculated using CDK (v2.5) [67] and the MinMax kernel [56]. The MS^2 matching scores and predicted ROs were used to compute max-marginal ranking scores using the framework by Bach et al. [34]. We used the author's implementation in version 0.2.3 [68]. The hyper-parameters β and k of the model were optimized for each evaluation LC-MS^2 experiment separately using the respective training data. To estimate β we generated 25 LC-MS^2 experiments from the training data and selected the β that maximized the Top20AUC [34] ranking performance. The sigmoid parameter k was estimated using Platt's method [69] calibrated using RankSVM's training data. We used 128 random spanning trees per evaluation LC-MS^2 experiment to compute the averaged max-marginals.

For the experiments comparing the different methods we used all LC-MS^2 experiments generated, except the ones from the MB-subsets "CE_001", "ET_002", "KW_000" and "RP_000" (see Extended Data Table 1). For those subsets the evaluation LC-MS^2 experiment contain all available (MS^2, RT)-tuples, leaving no LC system specific data to train the RT (MS^2+RT) or log$P$ (MS^2+log$P$) prediction models. The RT and log$P$ prediction models are trained in a structure disjoint fashion using the RT data of the particular MB-subset associated with the evaluation LC-MS^2. The RO prediction model used by MS^2+RO is trained structure disjoint as well, but using the RTs of all MB-subsets.

Data availability

All data used in our experiments is available online (`https://zenodo.org/record/5854661`). The candidate rankings of all LC-MS^2 experiments are available online: ALLDATA (`https://zenodo.org/record/6036208`)) and ONLYSTEREO (`https://zenodo.org/record/6037629`).

Code availability

The source code developed for this study is available on GitHub: Structure Support Vector Machine (SSVM) implementation (`https://github.com/aalto-ics-kepaco/msms_rt_ssvm`); scripts to run the experiments (`https://github.com/aalto-ics-kepaco/lcms2struct_exp`); and, the library implementing the MassBank pre-processing (`https://github.com/bachi55/massbank2db`). The candidate fingerprints where computed by the ROSVM Python library [66] (v0.4.0, `https://github.com/bachi55/rosvm`) using the RDKit (2021.03.1) in the backend. The SSVM library uses the max-marginal inference solver implemented by Bach et al. [34] (v0.2.3, `https://github.com/aalto-ics-kepaco/msms_rt_score_integration`).

References

[1] Ricardo R. da Silva et al. "Illuminating the dark matter in metabolomics". In: *Proceedings of the National Academy of Sciences* 112.41 (2015), pp. 12549–12550. ISSN: 0027-8424. DOI: 10.1073/pnas.1516878112. eprint: https://www.pnas.org/content/112/41/12549.full.pdf. URL: https://www.pnas.org/content/112/41/12549.

[2] Alexander A. Aksenov et al. "Global chemical analysis of biology by mass spectrometry". In: *Nature Reviews Chemistry* 1.7 (2017), p. 0054. ISSN: 2397-3358. DOI: 10.1038/s41570-017-0054. URL: https://doi.org/10.1038/s41570-017-0054.

[3] Ivana Blaženović et al. "Structure annotation of all mass spectra in untargeted metabolomics". In: *Analytical chemistry* 91.3 (2019), pp. 2155–2162.

[4] Ivana Blaženović et al. "Software Tools and Approaches for Compound Identification of LC-MS/MS Data in Metabolomics". In: *Metabolites* 8.2 (2018). ISSN: 2218-1989. DOI: 10.3390/metabo8020031. URL: https://www.mdpi.com/2218-1989/8/2/31.

[5] Emma L. Schymanski et al. "Critical Assessment of Small Molecule Identification 2016: automated methods". In: *Journal of Cheminformatics* 9.1 (Mar. 2017), p. 22. ISSN: 1758-2946. DOI: 10.1186/s13321-017-0207-1. URL: https://doi.org/10.1186/s13321-017-0207-1.

[6] Dai Hai Nguyen et al. "Recent advances and prospects of computational methods for metabolite identification: a review with emphasis on machine learning approaches". In: *Briefings in Bioinformatics* 20.6 (Aug. 2018), pp. 2028–2043. ISSN: 1477-4054. DOI: 10.1093/bib/bby066. eprint: https://academic.oup.com/bib/article-pdf/20/6/2028/31789414/bby066.pdf. URL: https://doi.org/10.1093/bib/bby066.

[7] Sebastian Wolf et al. "In silico fragmentation for computer assisted identification of metabolite mass spectra". In: *BMC Bioinformatics* 11.1 (2010), pp. 1–12. ISSN: 1471-2105.

[8] Kai Dührkop et al. "Searching molecular structure databases with tandem mass spectra using CSI:FingerID". In: *Proceedings of the National Academy of Sciences (PNAS)* (2015). eprint: http://www.pnas.org/content/early/2015/09/16/1509788112.full.pdf. URL: http://www.pnas.org/content/early/2015/09/16/1509788112.abstract.

[9] Felicity Allen et al. "Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification". In: *Metabolomics* 11.1 (2015), pp. 98–110. ISSN: 1573-3882. URL: http://dx.doi.org/10.1007/s11306-014-0676-4.

[10] Céline Brouard et al. "Fast metabolite identification with Input Output Kernel Regression". In: *Bioinformatics* 32.12 (2016), pp. i28–i36.

[11] Christoph Ruttkies et al. "MetFrag relaunched: incorporating strategies beyond in silico fragmentation". In: *Journal of Cheminformatics* 8.1 (Jan. 2016), p. 3. ISSN: 1758-2946. DOI: 10.1186/s13321-016-0115-9. URL: https://doi.org/10.1186/s13321-016-0115-9.

[12] Céline Brouard et al. "Magnitude-Preserving Ranking for Structured Outputs". In: *Proceedings of the Ninth Asian Conference on Machine Learning*. Ed. by Min-Ling Zhang et al. Vol. 77. Proceedings of Machine Learning Research. PMLR, 15–17 Nov 2017, pp. 407–422. URL: http://proceedings.mlr.press/v77/brouard17a.html.

[13] Dai Hai Nguyen et al. "SIMPLE: Sparse Interaction Model over Peaks of moLEcules for fast, interpretable metabolite identification from tandem mass spectra". In: *Bioinformatics* 34.13 (2018), pp. i323–i332. DOI: 10.1093/bioinformatics/bty252. eprint: /oup/backfile/content_public/journal/bioinformatics/34/13/10.1093_bioinformatics_bty252/1/bty252.pdf. URL: http://dx.doi.org/10.1093/bioinformatics/bty252.

[14] Yuanyue Li et al. "Identification of metabolites from tandem mass spectra with a machine learning approach utilizing structural features". In: *Bioinformatics* 36.4 (Oct. 2019), pp. 1213–1218. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz736. URL: https://doi.org/10.1093/bioinformatics/btz736.

[15] Christoph Ruttkies et al. "Improving MetFrag with statistical learning of fragment annotations". In: *BMC bioinformatics* 20.1 (2019), p. 376.

[16] Dai Hai Nguyen et al. "ADAPTIVE: leArning DAta-dePendenT, concIse molecular VEctors for fast, accurate metabolite identification from tandem mass spectra". In: *Bioinformatics* 35.14 (July 2019), pp. i164–i172. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz319. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/14/i164/28913118/btz319.pdf. URL: https://doi.org/10.1093/bioinformatics/btz319.

[17] Kai Dührkop et al. "SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information". In: *Nat Methods* (2019). Doi 10.1038/s41592-019-0344-8. DOI: 10.1038/s41592-019-0344-8.

[18] Fei Wang et al. "CFM-ID 4.0: More Accurate ESI-MS/MS Spectral Prediction and Compound Identification". In: *Analytical Chemistry* 0.0 (2021). PMID: 34403256, null. DOI: 10.1021/acs.analchem.1c01465. eprint: https://doi.org/10.1021/acs.analchem.1c01465. URL: https://doi.org/10.1021/acs.analchem.1c01465.

[19] David S Wishart et al. "HMDB 4.0: the human metabolome database for 2018". In: *Nucleic Acids Research* 46.D1 (Nov. 2017), pp. D608–D617. ISSN: 0305-1048. DOI: 10.1093/nar/gkx1089. eprint: https://academic.oup.com/nar/article-pdf/46/D1/D608/23162277/gkx1089.pdf. URL: https://doi.org/10.1093/nar/gkx1089.

[20] Sunghwan Kim et al. "PubChem in 2021: new data content and improved web interfaces". In: *Nucleic Acids Research* 49.D1 (Nov. 2020), pp. D1388–D1395. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa971. eprint: https://academic.oup.com/nar/article-pdf/49/D1/D1388/35363961/gkaa971.pdf. URL: https://doi.org/10.1093/nar/gkaa971.

[21] Jan Stanstrup et al. "PredRet: Prediction of Retention Time by Direct Mapping between Multiple Chromatographic Systems". In: *Analytical Chemistry* 87.18 (2015). PMID: 26289378, pp. 9421–9428. URL: http://dx.doi.org/10.1021/acs.analchem.5b02287.

[22] Dorrain Yanwen Low et al. "Data sharing in PredRet for accurate prediction of retention time: Application to plant food bioactive compounds". In: *Food Chemistry* 357 (2021), p. 129757. ISSN: 0308-8146. DOI: https://doi.org/10.1016/j.foodchem.2021.129757. URL: https://www.sciencedirect.com/science/article/pii/S0308814621007639.

[23] S. Fanali et al. *Liquid Chromatography: Fundamentals and Instrumentation*. Handbooks in Separation Science. Elsevier Science, 2013. ISBN: 9780124158672.

[24] Michael Witting et al. "Current status of retention time prediction in metabolite identification". In: *Journal of Separation Science* 43.9-10 (2020), pp. 1746–1754. DOI: 10.1002/jssc.202000060. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jssc.202000060. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jssc.202000060.

[25] Robbin Bouwmeester et al. "Comprehensive and empirical evaluation of machine learning algorithms for small molecule LC retention time prediction". In: *Analytical chemistry* 91.5 (2019), pp. 3694–3703.

[26] Fabian Aicheler et al. "Retention Time Prediction Improves Identification in Nontargeted Lipidomics Approaches". In: *Analytical chemistry* 87.15 (2015), pp. 7698–7704.

[27] Milinda A Samaraweera et al. "Evaluation of an Artificial Neural Network Retention Index Model for Chemical Structure Identification in Nontargeted Metabolomics". In: *Analytical chemistry* 90.21 (2018), pp. 12752–12760.

[28] Paolo Bonini et al. "Retip: Retention Time Prediction for Compound Annotation in Untargeted Metabolomics". In: *Analytical Chemistry* 0.0 (2020). PMID: 32390414, null. DOI: 10.1021/acs.analchem.9b05765. eprint: https://doi.org/10.1021/acs.analchem.9b05765. URL: https://doi.org/10.1021/acs.analchem.9b05765.

[29] Qiong Yang et al. "Prediction of Liquid Chromatographic Retention Time with Graph Neural Networks to Assist in Small Molecule Identification". In: *Anal. Chem.* (Jan. 2021). ISSN: 0003-2700. DOI: 10.1021/acs.analchem.0c04071. URL: https://doi.org/10.1021/acs.analchem.0c04071.

[30] Robbin Bouwmeester et al. "Generalized Calibration Across Liquid Chromatography Setups for Generic Prediction of Small-Molecule Retention Times". In: *Analytical Chemistry* 92.9 (2020). PMID: 32281370, pp. 6571–6578. DOI: 10.1021/acs.analchem.0c00233. eprint: https://doi.org/10.1021/acs.analchem.0c00233. URL: https://doi.org/10.1021/acs.analchem.0c00233.

[31] Eric Bach et al. "Liquid-chromatography retention order prediction for metabolite identification". In: *Bioinformatics* 34.17 (2018), pp. i875–i883. DOI: 10.1093/bioinformatics/bty590. eprint: /oup/backfile/content_public/journal/bioinformatics/34/17/10.1093_bioinformatics_bty590/2/bty590.pdf. URL: http://dx.doi.org/10.1093/bioinformatics/bty590.

[32] J Jay Liu et al. “Quantitative Structure–Retention Relationships with Non-Linear Programming for Prediction of Chromatographic Elution Order”. In: *International journal of molecular sciences* 20.14 (2019), p. 3443.

[33] Petar Žuvela et al. “Prediction of Chromatographic Elution Order of Analytical Mixtures Based on Quantitative Structure-Retention Relationships and Multi-Objective Optimization”. In: *Molecules* 25.13 (2020), p. 3085.

[34] Eric Bach et al. “Probabilistic framework for integration of mass spectrum and retention time information in small molecule identification”. In: *Bioinformatics* (Nov. 2020). btaa998. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa998. eprint: https://academic.oup.com/bioinformatics/advance-article-pdf/doi/10.1093/bioinformatics/btaa998/34899743/btaa998.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa998.

[35] I Tsochantaridis et al. “Large margin methods for structured and interdependent output variables”. In: *Journal of Machine Learning Research (JMLR)* 6 (2005).

[36] Ben Taskar et al. “Max-Margin Markov Networks”. In: *Advances in Neural Information Processing Systems 16*. Ed. by S. Thrun et al. MIT Press, 2004, pp. 25–32. URL: http://papers.nips.cc/paper/2397-max-margin-markov-networks.pdf.

[37] Hisayuki Horai et al. “MassBank: a public repository for sharing mass spectral data for life sciences”. In: *Journal of mass spectrometry* 45.7 (2010), pp. 703–714.

[38] Harry Pence et al. “ChemSpider: An Online Chemical Information Resource”. In: *Journal of Chemical Education* 87 (Aug. 2010). DOI: 10.1021/ed100697w.

[39] Emma Louise Schymanski et al. “Empowering Large Chemical Knowledge Bases for Exposomics: Pubchemlite Meets Metfrag”. In: *Journal of Cheminformatics* (2021). ISSN: 2693-5015. URL: https://doi.org/10.21203/rs.3.rs-107432/v1.

[40] Andreas Schüller et al. “SMILIB: Rapid Assembly of Combinatorial Libraries in SMILES Notation”. In: *QSAR & Combinatorial Science* 22.7 (2003), pp. 719–721. DOI: https://doi.org/10.1002/qsar.200310008. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qsar.200310008. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qsar.200310008.

[41] Andreas Schüller et al. “SmiLib v2.0: A Java-Based Tool for Rapid Combinatorial Library Enumeration”. In: *QSAR & Combinatorial Science* 26.3 (2007), pp. 407–410. DOI: https://doi.org/10.1002/qsar.200630101. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qsar.200630101. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qsar.200630101.

[42] Martin Wainwright et al. “Tree consistency and bounds on the performance of the max-product algorithm and its generalizations”. In: *Statistics and Computing* 14.2 (Apr. 2004), pp. 143–166. ISSN: 1573-1375. DOI: 10.1023/B:STCO.0000021412.33763.d5. URL: https://doi.org/10.1023/B:STCO.0000021412.33763.d5.

[43] David J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2005.

[44] Patrick Pletscher et al. “Spanning Tree Approximations for Conditional Random Fields”. In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk et al. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 16–18 Apr 2009, pp. 408–415. URL: http://proceedings.mlr.press/v5/pletscher09a.html.

[45] Hongyu Su et al. “Multilabel classification through random graph ensembles”. In: *Machine Learning* 99.2 (2015), pp. 231–256. ISSN: 1573-0565. DOI: 10.1007/s10994-014-5465-9. URL: https://doi.org/10.1007/s10994-014-5465-9.

[46] Juho Rousu et al. “Kernel-based learning of hierarchical multilabel classification models”. In: *Journal of Machine Learning Research* 7.Jul (2006), pp. 1601–1626.

[47] Eric Bach. *massbank2db: Build a machine learning ready SQLite database from MassBank.* Version 0.9.0. Jan. 2022. URL: `https://github.com/bachi55/massbank2db`.

[48] André Elisseeff et al. "A kernel method for multi-labelled classification". In: *Advances in neural information processing systems.* 2002, pp. 681–687.

[49] Thorsten Joachims. "Optimizing Search Engines Using Clickthrough Data". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '02. Edmonton, Alberta, Canada: ACM, 2002, pp. 133–142. ISBN: 1-58113-567-X. DOI: `10.1145/775047.775067`. URL: `http://doi.acm.org/10.1145/775047.775067`.

[50] Tiejun Cheng et al. "Computation of Octanol-Water Partition Coefficients by Guiding an Additive Model with Knowledge". In: *Journal of Chemical Information and Modeling* 47.6 (2007). PMID: 17985865, pp. 2140–2148. DOI: `10.1021/ci700257y`. eprint: `https://doi.org/10.1021/ci700257y`. URL: `https://doi.org/10.1021/ci700257y`.

[51] Yannick Djoumbou Feunang et al. "ClassyFire: automated chemical classification with a comprehensive, computable taxonomy". In: *Journal of cheminformatics* 8.1 (2016), p. 61.

[52] Thomas Gärtner et al. "On structured output training: hard cases and an efficient alternative". In: *Machine Learning* 76.2 (2009), pp. 227–242. ISSN: 1573-0565. DOI: `10.1007/s10994-009-5129-3`. URL: `https://doi.org/10.1007/s10994-009-5129-3`.

[53] Yexiang Xue et al. "Solving Marginal MAP Problems with NP Oracles and Parity Constraints". In: *Advances in Neural Information Processing Systems.* Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: `https://proceedings.neurips.cc/paper/2016/file/a532400ed62e772b9dc0b86f46e583ff-Paper.pdf`.

[54] Simon Lacoste-Julien et al. "Block-coordinate Frank-Wolfe optimization for structural SVMs". In: *International Conference on Machine Learning.* PMLR. 2013, pp. 53–61.

[55] Marguerite Frank et al. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110. DOI: `10.1002/nav.3800030109`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800030109`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109`.

[56] Liva Ralaivola et al. "Graph kernels for chemical informatics". In: *Neural networks* 18.8 (2005), pp. 1093–1110.

[57] Stephen R. Heller et al. "InChI, the IUPAC International Chemical Identifier". In: *Journal of Cheminformatics* 7.1 (2015), p. 23. ISSN: 1758-2946. DOI: `10.1186/s13321-015-0068-4`. URL: `https://doi.org/10.1186/s13321-015-0068-4`.

[58] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36. DOI: `10.1021/ci00057a005`. eprint: `https://pubs.acs.org/doi/pdf/10.1021/ci00057a005`. URL: `https://pubs.acs.org/doi/abs/10.1021/ci00057a005`.

[59] Jeramie Watrous et al. "Mass spectral molecular networking of living microbial colonies". In: *Proceedings of the National Academy of Sciences* 109.26 (2012), E1743–E1752. ISSN: 0027-8424. DOI: `10.1073/pnas.1203689109`. eprint: `https://www.pnas.org/content/109/26/E1743.full.pdf`. URL: `https://www.pnas.org/content/109/26/E1743`.

[60] Florian Huber et al. "matchms - processing and similarity evaluation of mass spectrometry data." In: *Journal of Open Source Software* 5.52 (2020), p. 2411. DOI: `10.21105/joss.02411`. URL: `https://doi.org/10.21105/joss.02411`.

[61] David Rogers et al. "Extended-Connectivity Fingerprints". In: *Journal of Chemical Information and Modeling* 50.5 (2010). PMID: 20426451, pp. 742–754. DOI: `10.1021/ci100050t`. eprint: `http://dx.doi.org/10.1021/ci100050t`. URL: `http://dx.doi.org/10.1021/ci100050t`.

[62] John W. Dolan. *Column Dead Time as a Diagnostic Tool*. Tech. rep. 1. Jan. 2014, pp. 24–29. URL: `http://www.chromatographyonline.com/column-dead-time-diagnostic-tool`.

[63] Kalervo Järvelin et al. "Cumulated Gain-Based Evaluation of IR Techniques". In: *ACM Trans. Inf. Syst.* 20.4 (Oct. 2002), pp. 422–446. ISSN: 1046-8188. DOI: `10.1145/582415.582418`. URL: `https://doi.org/10.1145/582415.582418`.

[64] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[65] Harris Drucker et al. "Support vector regression machines". In: *Advances in neural information processing systems*. 1997, pp. 155–161.

[66] Eric Bach. *Retention Order Support Vector Machine (ROSVM)*. Version 0.4.0. Nov. 2021. URL: `https://github.com/bachi55/rosvm`.

[67] Egon L. Willighagen et al. "The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching". In: *Journal of Cheminformatics* 9.1 (June 2017), p. 33. ISSN: 1758-2946. DOI: `10.1186/s13321-017-0220-4`. URL: `https://doi.org/10.1186/s13321-017-0220-4`.

[68] Eric Bach. *msmsrt_scorer: Probabilistic framework for integration of mass spectrum and retention order information*. Version 0.2.3. Nov. 2021. URL: `https://github.com/aalto-ics-kepaco/msms_rt_score_integration`.

[69] John Platt. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* 10.3 (June 2000).

Acknowledgments

The work by E.B. and J.R. was in part supported by Academy of Finland grants 310107 (MACOME) and 334790 (MAGITICS). E.L.S. acknowledges funding support from the Luxembourg National Research Fund (FNR) for project A18/BM/12341006 and discussions with Drs. Greg Landrum (ETHZ) and Evan Bolton (NCBI/NLM/NIH). The authors acknowledge CSC – IT Center for Science, Finland and Aalto Science-IT infrastructure, Finland for generous computational resources. E.B. thanks Dr. K. Dührkop for generating the SIRIUS candidate sets and predicting the SIRIUS MS^2 scores.

Authors contributions

E.B. and J.R. designed the research. E.B. implemented the MassBank pre-processing. E.B. developed, implemented and evaluated the computational method. E.B., E.L.S. and J.R. interpreted the results. E.B., E.L.S. and J.R. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

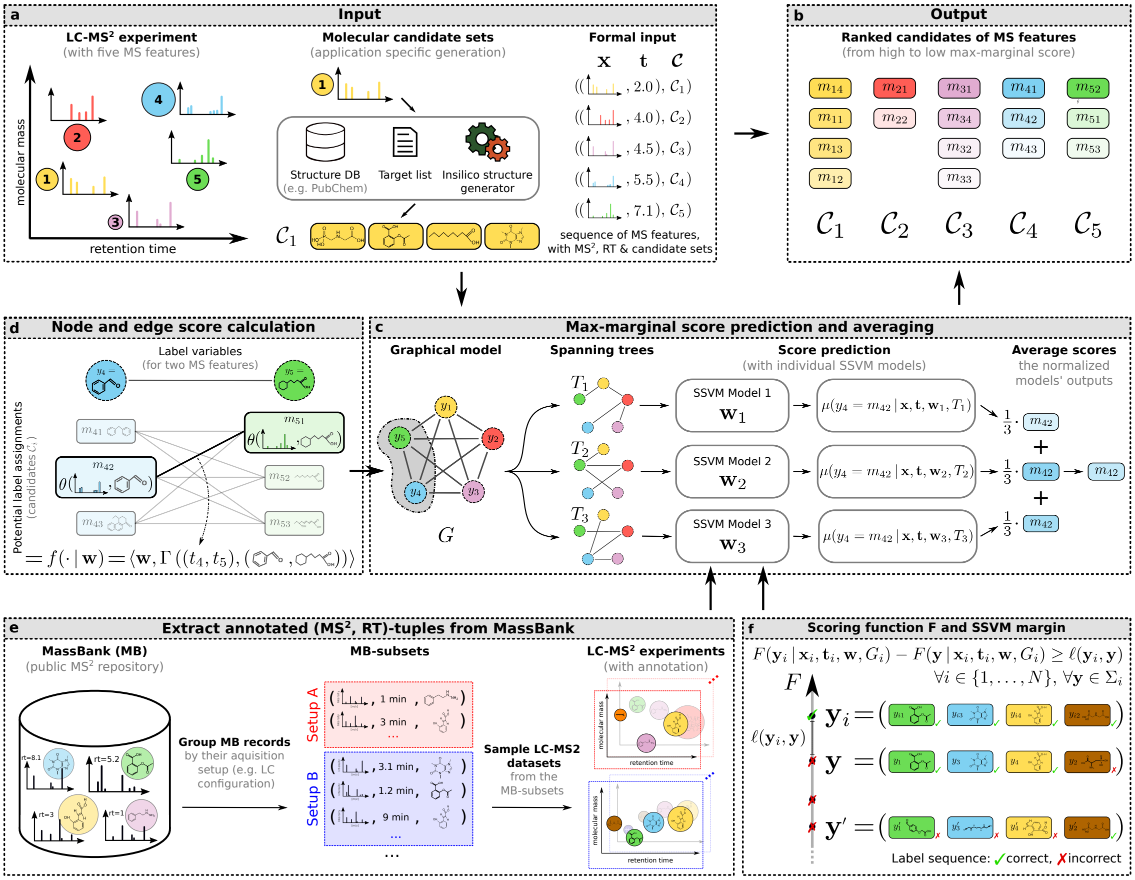
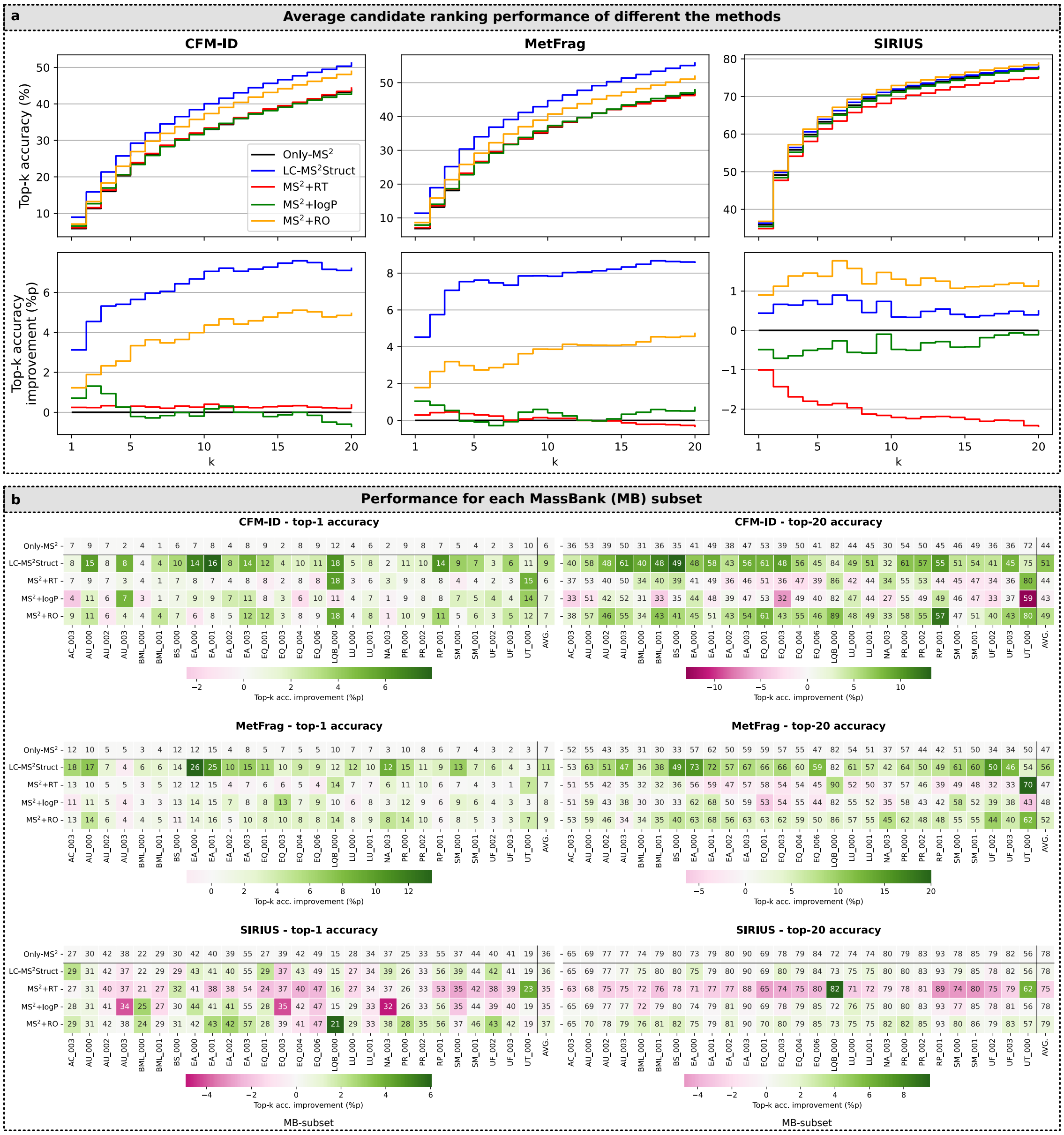

a
Input
LC-MS² experiment
(with five MS features)
molecular mass
retention time
1
2
3
4
5
Molecular candidate sets
(application specific generation)
Structure DB
(e.g. PubChem)
Target list
Insilico structure generator
$\mathcal{C}_1$
Formal input
$\mathbf{x}$ $\mathbf{t}$ $\boldsymbol{\mathcal{C}}$
$((\ , 2.0), \mathcal{C}_1)$
$((\ , 4.0), \mathcal{C}_2)$
$((\ , 4.5), \mathcal{C}_3)$
$((\ , 5.5), \mathcal{C}_4)$
$((\ , 7.1), \mathcal{C}_5)$
sequence of MS features, with MS², RT & candidate sets
b
Output
Ranked candidates of MS features
(from high to low max-marginal score)
m_{14} m_{21} m_{31} m_{41} m_{52}
m_{11} m_{22} m_{34} m_{42} m_{51}
m_{13} m_{32} m_{43} m_{53}
m_{12} m_{33}
$\mathcal{C}_1$ $\mathcal{C}_2$ $\mathcal{C}_3$ $\mathcal{C}_4$ $\mathcal{C}_5$
d
Node and edge score calculation
Label variables
(for two MS features)
$y_4 =$
$y_5 =$
Potential label assignments
(candidates $\mathcal{C}_i$)
m_{41}
m_{42}
$\theta($
m_{43}
m_{51}
m_{52}
m_{53}
$= f(\cdot \mid \mathbf{w}) = \langle \mathbf{w}, \Gamma((t_4, t_5), (\ , \ )) \rangle$
c
Max-marginal score prediction and averaging
Graphical model
y_1 y_2 y_3 y_4 y_5
G
Spanning trees
T_1
T_2
T_3
Score prediction
(with individual SSVM models)
SSVM Model 1 $\mathbf{w}_1$
SSVM Model 2 $\mathbf{w}_2$
SSVM Model 3 $\mathbf{w}_3$
$\mu(y_4 = m_{42} \mid \mathbf{x}, \mathbf{t}, \mathbf{w}_1, T_1)$
$\mu(y_4 = m_{42} \mid \mathbf{x}, \mathbf{t}, \mathbf{w}_2, T_2)$
$\mu(y_4 = m_{42} \mid \mathbf{x}, \mathbf{t}, \mathbf{w}_3, T_3)$
Average scores
the normalized models' outputs
$\frac{1}{3} \cdot m_{42}$
+
$\frac{1}{3} \cdot m_{42}$
+
$\frac{1}{3} \cdot m_{42}$
m_{42}
e
Extract annotated (MS², RT)-tuples from MassBank
MassBank (MB)
(public MS² repository)
rt=8.1
rt=5.2
rt=3
rt=1
Group MB records
by their aquisition setup (e.g. LC configuration)
MB-subsets
Setup A
1 min
3 min
...
Setup B
3.1 min
1.2 min
9 min
...
...
Sample LC-MS2 datasets
from the MB-subsets
LC-MS² experiments
(with annotation)
molecular mass
retention time
f
Scoring function F and SSVM margin
$F(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{t}_i, \mathbf{w}, G_i) - F(\mathbf{y} \mid \mathbf{x}_i, \mathbf{t}_i, \mathbf{w}, G_i) \geq \ell(\mathbf{y}_i, \mathbf{y})$
$\forall i \in \{1, \ldots, N\}, \forall \mathbf{y} \in \Sigma_i$
F
$\ell(\mathbf{y}_i, \mathbf{y})$
$\mathbf{y}_i = (y_{i1}, y_{i3}, y_{i4}, y_{i2})$
$\mathbf{y} = (y_1, y_3, y_4, y_2)$
$\mathbf{y}' = (y_1', y_3', y_4', y_2')$
Label sequence: ✓correct, ✗incorrect

Figure 1: **Overview of the LC-MS²Struct workflow.** **a**: Input to LC-MS²Struct during the application phase. The LC-MS² experiment results in a set of (MS², RT)-tuples. The MS information is used to generate a molecular candidate set for each MS feature. **b**: Output of LC-MS²Struct are the ranked molecular candidates for each MS feature. **c**: A fully connected graph G models the pairwise dependency between the MS features. Using a set of random spanning trees T_k and Structured Support Vector Machines (SSVM) we predict the max-marginal scores for each candidate used for the ranking. **d**: The MS² and RO information is used to scores the nodes and edges in the graph G. **e**: To train the SSVM models and evaluate LC-MS²Struct, we extract MS² spectra and RTs from MassBank. We group the MassBank records such that their experimental setups are matching and simulate LC-MS² experiment. **f**: Main objective optimized during the training of the SSVM.



CFM-ID - top-1 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	7	9	7	2	4	1	6	7	8	4	8	9	2	8	9	12	4	6	2	9	8	7	5	4	2	3	10	6
LC-MS^{2}Struct	8	15	8	8	4	4	10	14	16	8	14	12	4	10	11	18	5	8	2	11	10	14	9	7	3	6	11	9
MS^2+RT	7	9	7	3	4	1	7	8	7	4	8	8	2	8	8	18	3	6	3	9	8	8	4	4	2	3	15	6
MS^2+logP	4	11	6	7	3	1	7	9	9	7	11	8	3	6	10	11	4	7	1	9	8	8	7	5	4	4	14	7
MS^2+RO	9	11	6	4	4	4	7	6	8	5	12	12	3	8	9	18	4	8	1	10	9	11	5	6	3	5	12	7

CFM-ID - top-20 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	36	53	39	50	31	36	35	41	49	38	47	53	39	50	41	82	44	45	30	54	50	45	46	49	36	36	72	44
LC-MS^{2}Struct	40	58	48	61	40	48	49	48	58	43	56	61	48	56	45	84	49	51	32	61	57	55	51	54	41	45	75	51
MS^2+RT	37	53	40	50	34	40	39	41	49	36	46	51	36	47	39	86	42	44	34	55	53	44	45	47	34	36	80	44
MS^2+logP	33	51	42	52	31	33	35	44	48	39	47	53	32	49	40	82	47	44	27	55	49	49	46	47	33	37	59	43
MS^2+RO	38	57	46	55	34	43	41	45	51	43	54	61	43	55	46	89	48	49	33	58	55	57	47	51	40	43	80	49

MetFrag - top-1 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	12	10	5	5	3	4	12	12	15	4	8	5	7	5	5	10	7	7	3	10	8	6	7	4	3	2	3	7
LC-MS^{2}Struct	18	17	7	4	6	6	14	26	25	10	15	11	10	9	9	12	12	10	12	15	11	9	13	7	6	4	3	11
MS^2+RT	13	10	5	5	3	5	12	12	15	4	7	6	6	5	4	14	7	7	6	11	10	6	7	4	3	1	7	7
MS^2+logP	11	11	5	4	3	3	13	14	15	7	8	8	13	7	9	10	6	8	3	12	9	6	9	6	4	3	3	8
MS^2+RO	13	14	6	4	4	5	11	14	16	5	10	8	10	8	8	14	8	9	8	14	10	6	8	5	3	3	7	9

MetFrag - top-20 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	52	55	43	35	31	30	33	57	62	50	59	59	57	55	47	82	54	51	37	57	44	42	51	50	34	34	50	47
LC-MS^{2}Struct	53	63	51	47	36	38	49	73	72	57	67	66	66	60	59	82	61	57	42	64	50	49	61	60	50	46	54	56
MS^2+RT	51	55	42	35	32	32	36	56	59	47	57	58	54	54	45	90	52	50	37	57	46	39	49	48	32	33	70	47
MS^2+logP	51	59	43	38	30	30	33	62	68	50	59	53	54	55	44	82	55	52	35	58	43	42	58	52	39	38	43	48
MS^2+RO	53	59	46	34	34	35	40	63	68	56	63	63	62	59	50	86	57	55	45	62	48	48	55	55	44	40	62	52

SIRIUS - top-1 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	27	30	42	38	22	29	30	42	40	39	55	27	39	42	49	15	28	34	37	25	33	55	37	44	40	41	19	36
LC-MS^{2}Struct	29	31	42	37	22	29	29	43	41	40	55	29	37	43	49	15	27	34	39	26	33	56	39	44	42	41	19	36
MS^2+RT	27	31	40	37	21	27	32	41	38	38	54	24	37	40	47	16	27	34	37	26	33	53	35	42	38	39	23	35
MS^2+logP	28	31	41	34	25	27	30	44	41	41	55	28	35	42	47	15	29	33	32	26	33	56	35	44	39	40	19	35
MS^2+RO	29	31	42	38	24	29	31	42	43	42	57	28	39	41	47	21	29	33	38	28	35	56	37	46	43	42	19	37

SIRIUS - top-20 accuracy

	AC_003	AU_000	AU_002	AU_003	BML_000	BML_001	BS_000	EA_000	EA_001	EA_002	EA_003	EQ_001	EQ_003	EQ_004	EQ_006	LQB_000	LU_000	LU_001	NA_003	PR_000	PR_002	RP_001	SM_000	SM_001	UF_002	UF_003	UT_000	AVG.
Only-MS^2	65	69	77	77	74	79	80	73	79	80	90	69	78	79	84	72	74	74	80	79	83	93	78	85	79	82	56	78
LC-MS^{2}Struct	65	69	77	77	75	80	80	75	79	80	90	69	80	79	84	73	75	75	80	80	83	93	79	85	78	82	56	78
MS^2+RT	63	68	75	75	72	76	78	71	77	77	88	65	74	75	80	82	71	72	79	78	81	89	74	80	75	79	62	75
MS^2+logP	65	69	77	77	72	79	80	74	79	81	90	69	78	79	85	72	76	75	80	80	83	93	77	85	78	81	56	78
MS^2+RO	65	70	78	79	76	81	82	75	79	81	90	70	80	79	85	73	75	75	82	82	85	93	80	86	79	83	57	79

Figure 2: **Different approaches to combine MS^2 and retention time (RT) information: a**: Comparison of the performance, measured by top-k accuracy, for the different ranking approaches combining MS^2 and RT information. The results shown are averaged accuracies over 350 sample MS feature sequences (LC-MS^2 experiments). **b**: Average top-k accuracies per MassBank (MB) subset rounded to full integers. The color encodes the performance improvement of each score integration method compared to Only-MS^2.

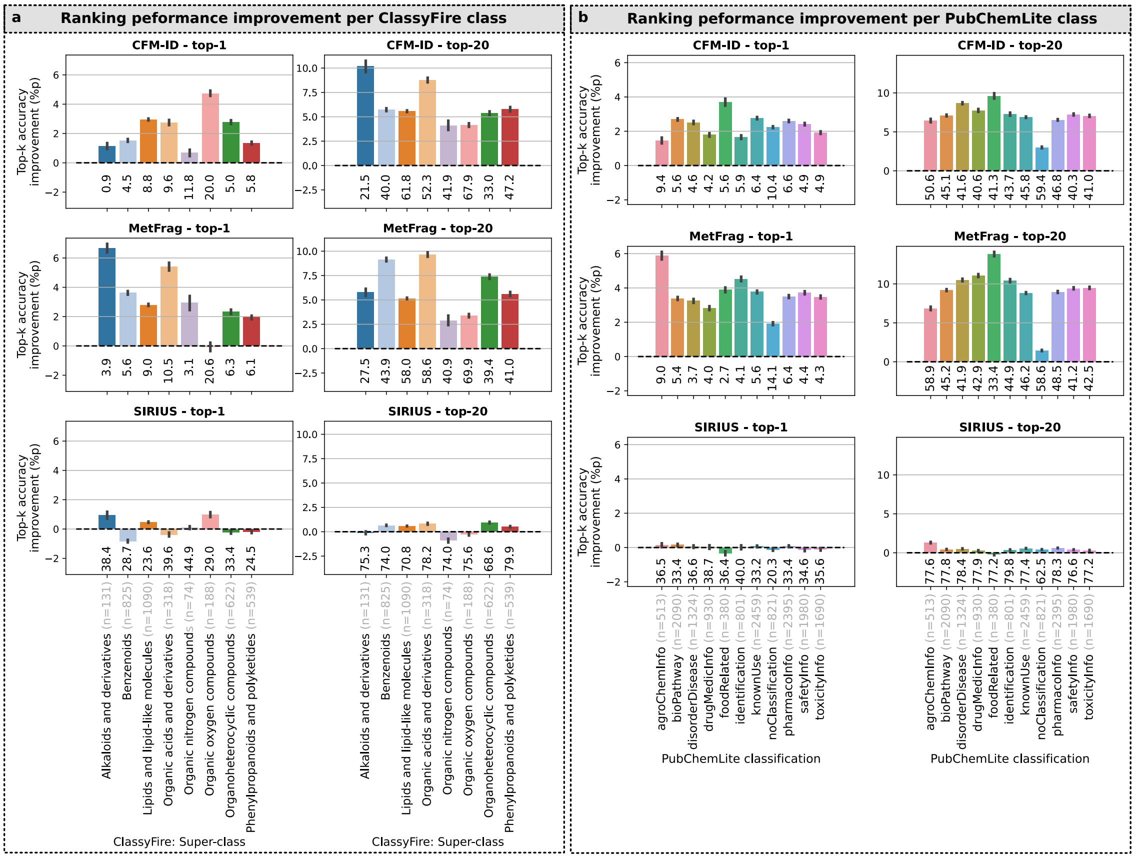


Figure 3: **Performance gain by LC-MS^{2}Struct across molecular classes.** The figure shows the average and 95%-confidence interval of the ranking performance (top-k) improvement of LC-MS^{2}Struct compared to Only-MS^2 (baseline). The top-k accuracies (%) under the bars show the Only-MS^2 performance. For each molecular class, the number of unique molecular structures in the class is denoted in the x-axis label (n). **a**: Molecular classification using the ClassyFire [51] framework. **b**: PubChemLite [39] annotation classification system. Molecules not present in PubChemLite are summarized under the “noClassification” category. Note that in PubChemLite a molecule can belong to multiple categories.

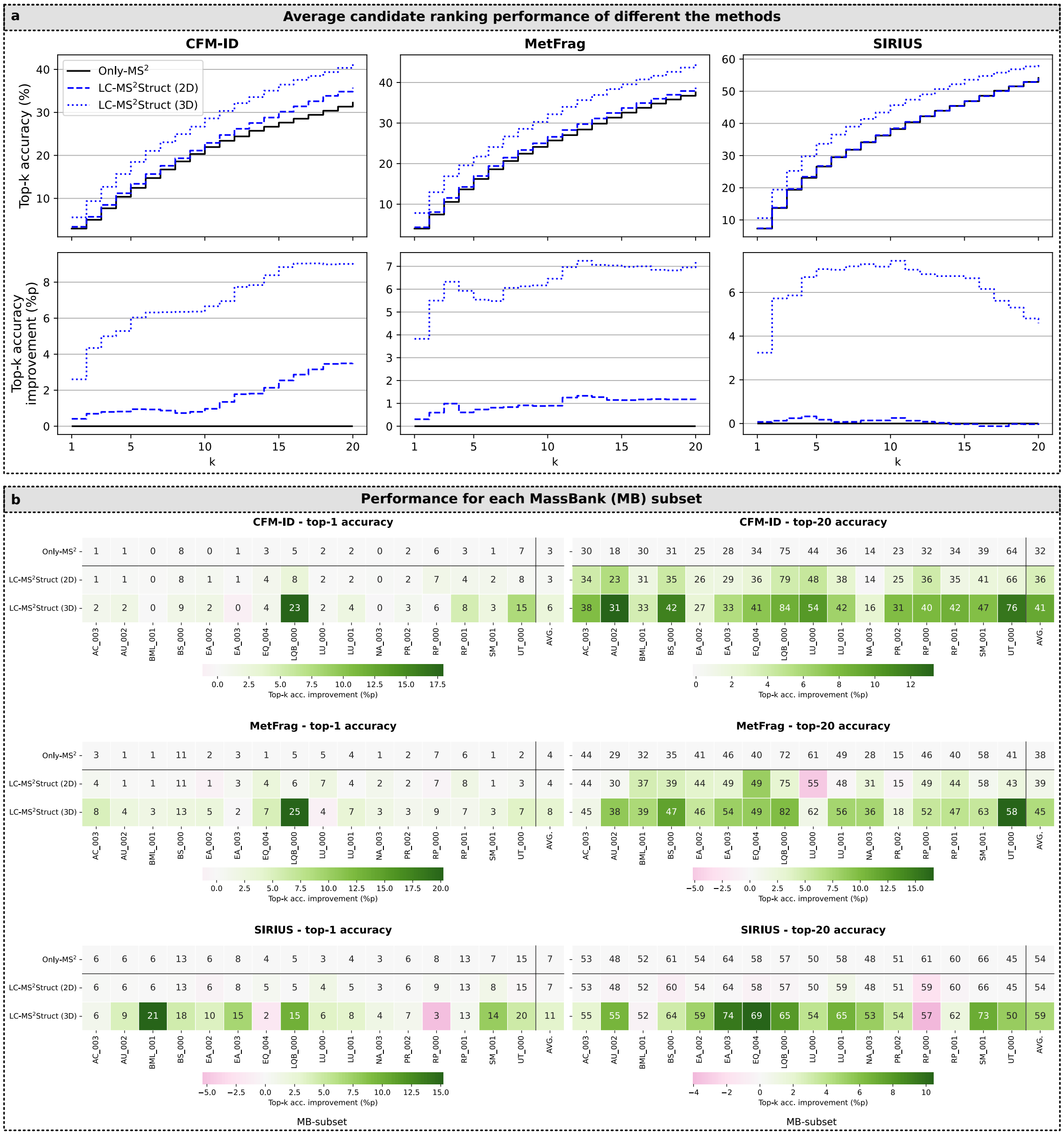


Figure 4: **Using LC-MS^{2}Struct with different feature representations. a**: Comparison of the performance, measured by top-k accuracy, of LC-MS^{2}Struct using either 2D (no stereochemistry) or 3D (with stereochemistry) molecular fingerprints. The results shown are averaged accuracies over 94 sample MS feature sequences (LC-MS^2 experiments). **b**: Average top-k accuracies per MassBank (MB) subset rounded to full integers. The color encodes the performance improvement of each score integration method compared to Only-MS^2.

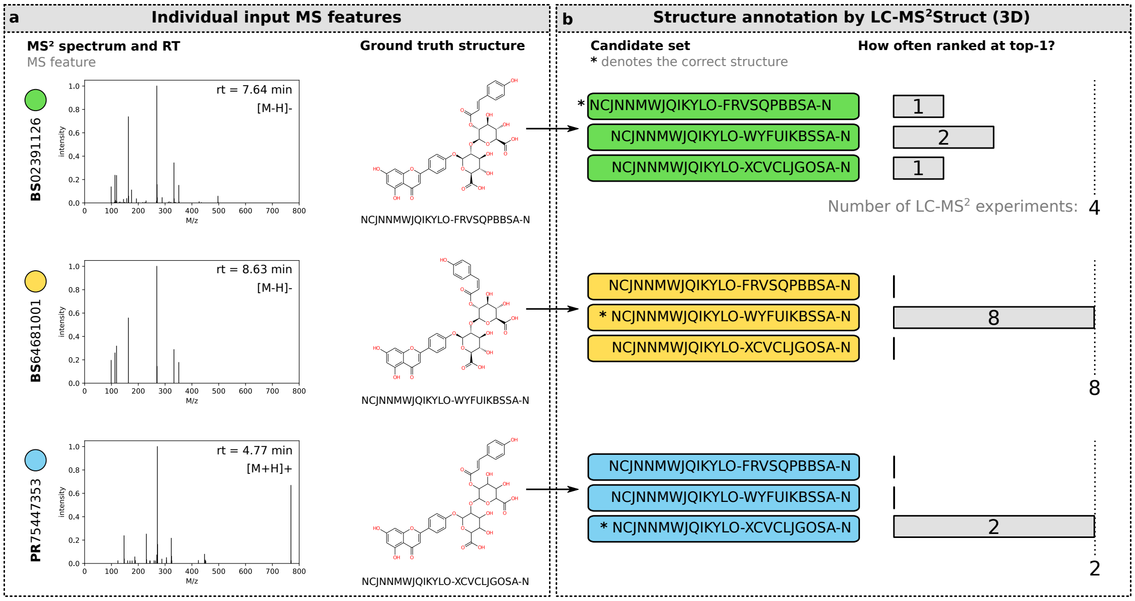


Figure 5: **Application of LC-MS^{2}Struct to annotate stereoisomers.** Post-hoc analysis of the stereoisomer annotation using LC-MS^{2}Struct for three (MS^2, RT)-tuples from our MassBank data associated with the same 2D skeleton (InChIKey first block). In our evaluation, all three MS features were analysed multiple times in different contexts (BS02391126 in 4, BS64681001 in 8 and PR75447353 in 2 LC-MS^2 experiments). **a**: MS features with their ground truth annotations. Two of the spectra (starting with BS) were measured under the same LC condition (MB-subset "BS_000"), demonstrating the separation of *E*/*Z*-isomers on LC columns. **b**: The candidate sets of the three features are identical (defined by the molecular formula $C_{36}H_{32}O_{19}$) and only contain three structures. For 12 out of the 14 LC-MS^2 experiments, LC-MS^{2}Struct predicts the correct *E*/*Z*-isomer.

Extended data figures and tables

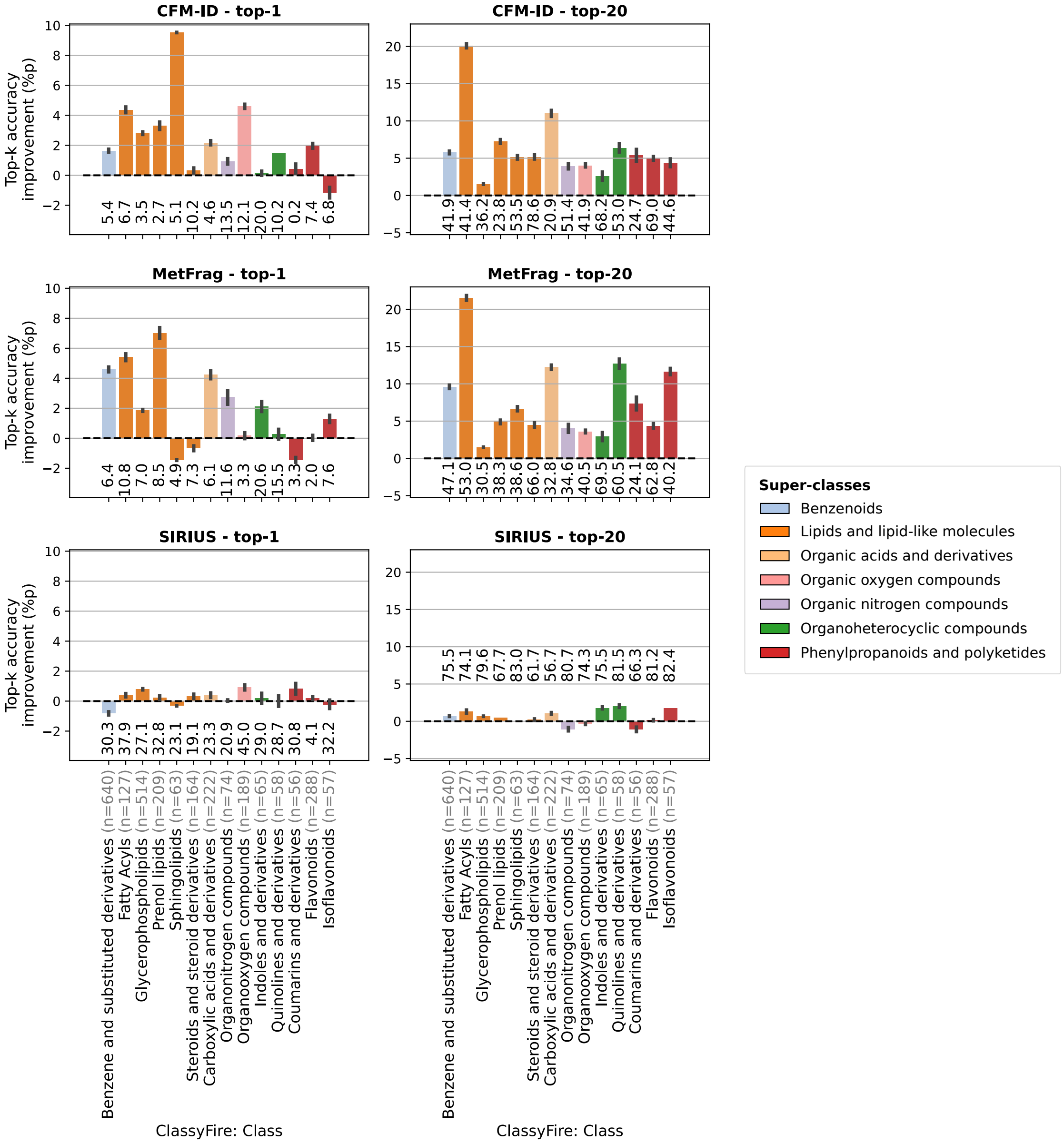


Figure 6: **Performance gain by LC-MS^{2}Struct across ClassyFire class-level annotations.** The figure shows the average and 95%-confidence interval of the ranking performance (top-k) improvement of LC-MS^{2}Struct compared to Only-MS^2 (baseline). The top-k accuracies (%) under the bars show the Only-MS^2 performance. For each molecule class, the number of unique molecular structures in the class is denoted in the x-axis label (n).

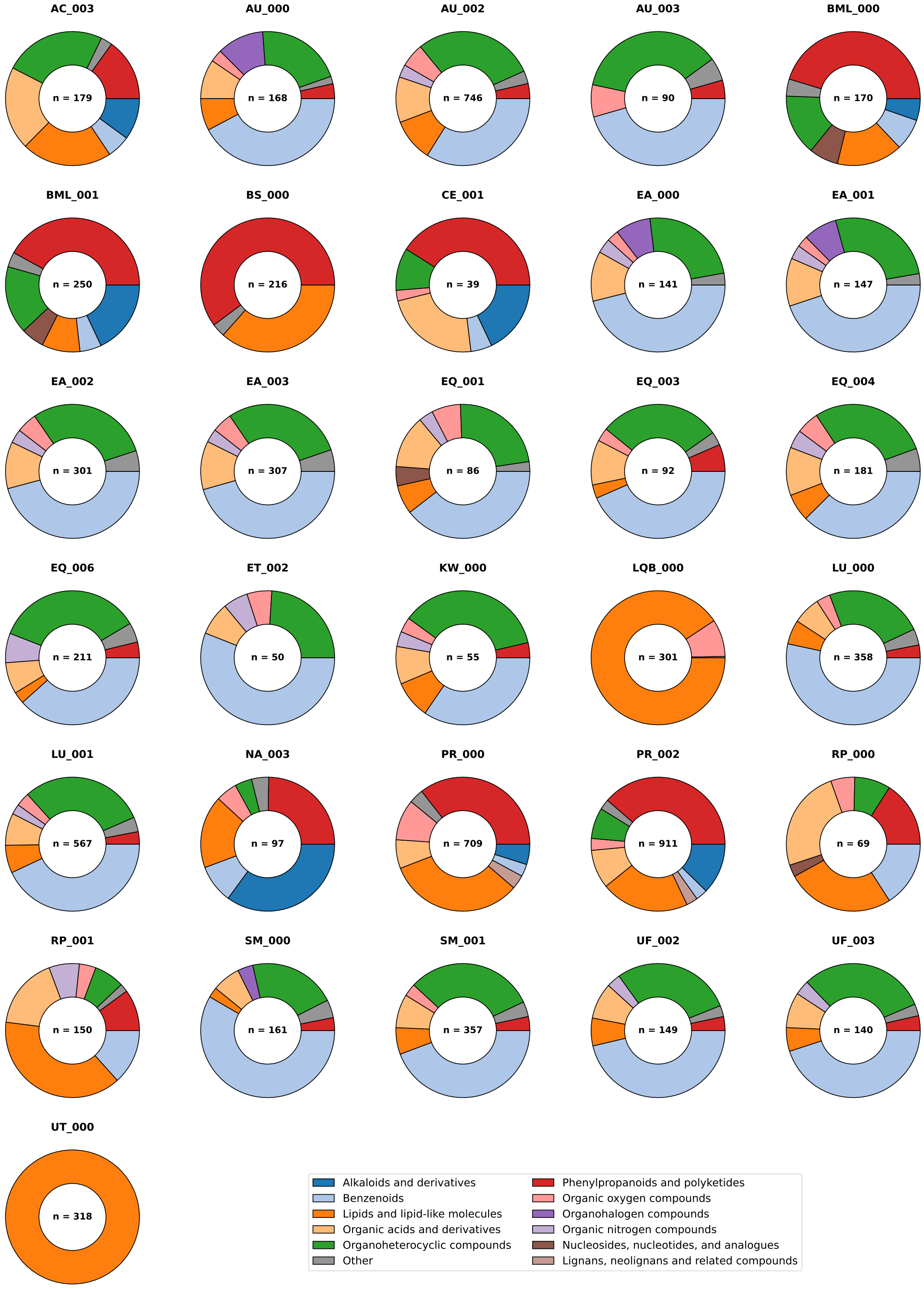
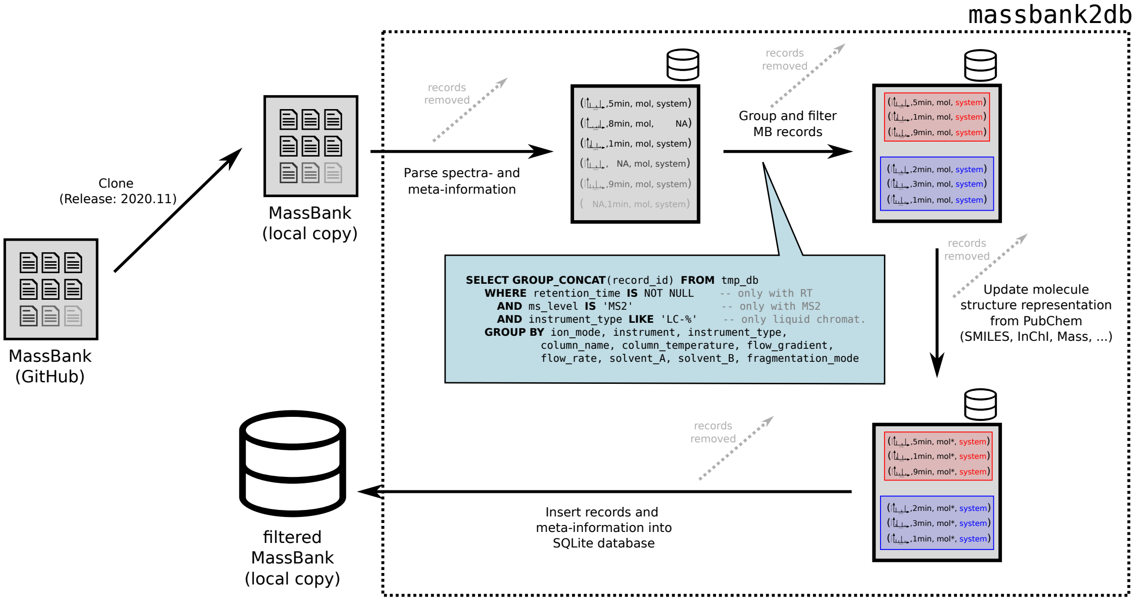


Figure 7: **Distribution of molecule classes in the MassBank (MB) subsets.** ClassyFire super-class distribution [51] for each MB-subset studied in our experiments. Within each MB-subset, the label "Other" is assigned to each super-class which makes up less then 2.5% of all molecules. The center label represents the number of examples for the respective MB-subset.



```sql
SELECT GROUP_CONCAT(record_id) FROM tmp_db
    WHERE retention_time IS NOT NULL     -- only with RT
      AND ms_level IS 'MS2'              -- only with MS2
      AND instrument_type LIKE 'LC-%'    -- only liquid chromat.
    GROUP BY ion_mode, instrument, instrument_type,
             column_name, column_temperature, flow_gradient,
             flow_rate, solvent_A, solvent_B, fragmentation_mode
```

Figure 8: **Processing pipeline of the MassBank data.** Illustration of the processing pipeline to extract the training data from MassBank. The depicted workflow is implemented in the "massbank2db" Python package [47].

Table 1: **Training and evaluation dataset sizes in our experiments.** We provide the number (#) of (MS^2, RT)-tuples used for the generation of training and evaluation LC-MS^2 experiments. For the ALLDATA setup the training and evaluation tuple-set is equal. The number of evaluation LC-MS^2 experiments depends on the number of available evaluation tuples.

	ALLDATA		**ONLYSTEREO**		
MB-subset	**#Tuples**	**#Exp.**	**#Tuples (train.)**	**#Tuples (eval.)**	**#Exp.**
AC_003	179	15	172	157	15
AU_000	168	15	146	23	-
AU_002	746	14	578	172	15
AU_003	90	15	77	21	-
BML_000	170	15	77	24	-
BML_001	250	15	125	33	1
BS_000	216	15	205	135	15
CE_001	39	1	30	19	-
EA_000	141	15	118	19	-
EA_001	147	15	126	19	-
EA_002	301	6	240	56	1
EA_003	307	6	246	57	1
EQ_001	86	15	68	28	-
EQ_003	92	15	64	6	-
EQ_004	181	15	127	51	1
EQ_006	211	15	138	15	-
ET_002	50	1	29	2	-
KW_000	55	1	43	4	-
LQB_000	301	6	271	270	5
LU_000	358	7	311	50	1
LU_001	567	11	472	101	15
NA_003	97	15	91	73	1
PR_000	709	14	131	21	-
PR_002	911	18	391	250	15
RP_000	69	1	55	35	1
RP_001	150	15	119	73	1
SM_000	161	15	136	12	-
SM_001	357	7	280	30	1
UF_002	149	15	124	18	-
UF_003	140	15	115	15	-
UT_000	318	6	294	293	5
Total	7716	354	5399	2082	94

Table 2: **Median candidate set size for the MassBank (MB) subsets.** The table shows the median number of molecular candidates per MB-subset used in our experiments. In the ALLDATA setup the candidates are identified by their InChIKey first block, where as for the Only-MS^2 setup the full InChIKey is used. The candidate number is computed based on the MB records which are used in the simulated LC-MS^2 experiments. For ONLYSTEREO, some MB-subsets are not used in the evaluation, and therefore their candidate set size is omitted (-).

MB-subset	**ALLDATA**	**ONLYSTEREO**
AC_003	305	384
AU_000	269	-
AU_002	1018.5	1434.5
AU_003	1297	-
BML_000	689	-
BML_001	1013.5	1688
BS_000	429	258
CE_001	819	-
EA_000	771	-
EA_001	570	-
EA_002	1373	1239
EA_003	1306	1097
EQ_001	425	-
EQ_003	759.5	-
EQ_004	872	1027
EQ_006	1045	-
ET_002	4957	-
KW_000	2010	-
LQB_000	73	106
LU_000	533	362.5
LU_001	998	751
NA_003	1024	1608
PR_000	109	-
PR_002	228	636
RP_000	760	1015
RP_001	658	723
SM_000	312	-
SM_001	800	1095.5
UF_002	1498	-
UF_003	1392.5	-
UT_000	56	93

Table 3: **MassBank (MB) information used to group the records.** Two MassBank records are considered to belong to the same MB-subset in our experiments, if all properties listed in the table are equal between them. See `https://github.com/MassBank/MassBank-web/blob/main/Documentation/MassBankRecordFormat.md` for a more comprehensive description of the MassBank records' fields.

Property	**Description**	**Example**
`contributor`	Contributor who uploaded a MassBank record	BGC_Munich
`accession prefix`	2-3 character long prefix further specifying the records of a contributor	EA, EQ
`instrument_type`	General type of instrument used for the LC-MS analysis	LC-ESI-QTOF
`ion_mode`	MS Ionization mode	negative
`instrument`	Commercial name and manufacturer of the MS instrument	Bruker maXis Impact
`fragmentation_mode`	Fragmentation method used for dissociation or fragmentation	CID
`column_name`	Commercial name and manufacturer of the LC instrument	Symmetry C18 Column, Waters
`column_temperature`	Static column temperature in LC-MS	40 C
`flow_gradient`	Gradient of mobile phases in LC-MS	0min:5%, 24min:95% (acetonitrile)
`flow_rate`	Flow Rate of liquid phase in LC	300 uL/min
`solvent_A`	Chemical composition of buffer solution (A)	H2O(0.1%HCOOH)
`solvent_B`	Chemical composition of buffer solution (B)	CH3CN(0.1%HCOOH)

Supplementary material

Table 4: **Meta-information for the MassBank (MB) subsets.** The the LC- and MS-conditions for each MB-subset.

THIS TABLE IS PROVIDED IN A SEPARATE FILE: `massbank_groups_meta_data.tsv`