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 authors: EB: eric.bach@aalto.fi; JR: juho.rousu@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 jointly predicts the annotations for a set of mass spectrometry features in a sample, using a novel structured prediction model trained to optimally combine the output of state-of-the-art MS^2 scorers and observed retention orders. We evaluate our method on a dataset covering all publicly available reversed phase LC-MS^2 data in the MassBank reference database, including 4327 molecules measured using 18 different LC conditions from 16 contributors, greatly expanding the chemical analytical space covered in previous multi-MS^2 scorer evaluations. LC-MS^{2}Struct obtains significantly higher annotation accuracy than earlier methods and improves the annotation accuracy of state-of-the-art MS^2 scorers by up to 106%. The use of stereochemistry-aware molecular fingerprints improves prediction performance, which highlights limitations in existing approaches and has strong implications for future computational LC-MS^2 developments.

Introduction

Structural annotation of small molecules in biological samples is a key bottleneck in various research fields including biomedicine, biotechnology, drug discovery and environmental sciences. Samples in untargeted metabolomics studies typically contain thousands of different molecules, the vast majority 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, is highly sensitive and applicable to a wide range of molecules. 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, they are separated according to their mass-to-charge ratio in a mass analyzer (MS^1). Finally, the molecular ions are isolated and fragmented in the tandem mass spectrometer (MS^2)

For each ion, the recorded fragments and their intensities constitute the MS^2 spectrum, containing information about the substructures in the molecule and serves as a basis for annotation efforts. In typical untargeted LC-MS^2 workflows, thousands of MS features (MS^1, MS^2, RT) arise from a single sample. The goal of 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 structural annotations for MS^2 spectra have been developed [7, 11–18, 8–10]. 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 such as HMDB [19] or PubChem [20] and subsequently

computing a match 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. Currently, even the best-of-class methods only reach an annotation accuracy of around 40% [9] in evaluations when searching large candidate sets such as those retrieved from PubChem. Therefore, in practice, a *ranked list* of molecular structures is provided to the user (*e.g.* top 20 structures). This level of performance is still a significant hindrance in metabolomics and other fields.

Interestingly, RT information remains underutilized in automated approaches for structure annotation based on MS^2, despite RTs being readily available in all LC-MS^2 pipelines and generally recognized as contributing valuable information [21, 22]. An explanation is that a molecule generally has different RTs under different LC conditions (mobile phase, column composition, etc.) [23, 24]. Typically, the RT information is used for post-processing of 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 based solely on the molecular structure of the candidate, and have been successfully applied to aid structure annotation [14, 26–29]. However, such prediction models generally have to be calibrated to the specific LC configuration [3], requiring at least some amount of target LC reference standard 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 systems). Therefore, RO predictors can be trained using a diverse set of RT reference data, and applied to out-of-dataset LC setups [31]. Integration of MS^2 and RO based scores using probabilistic graphical models improved the annotation performance in LC-MS^2 experiments [34].

Another somewhat neglected aspect in automated annotation pipelines is the treatment of stereochemistry, *i.e.* the different 3D variants of the molecules. The general assumption has been that LC-MS^2 data does not contain sufficient information to separate stereoisomers in samples [5, 24]. As a result, MS^2 scorers typically disregard the stereochemical information in the candidate structures and often output the same matching for different stereoisomers (*c.f.* [7, 9]). However, stereoisomers that vary in their double-bond orientation (*e.g.* *cis-trans* or *E-Z* isomerism) may have different shapes and thus exhibit different fragmentation and/or interactions with the LC system. Thus, ignoring stereochemistry in candidate processing may disregard LC-relevant stereochemical information. Furthermore, it is known that certain stereochemical configurations occur more frequently than others in nature and hence in the reference databases. Making use of such information can potentially improve annotation performance.

In this study we set out to provide a new perspective on jointly using MS^2 and RO combined with stereochemistry-aware molecular features for the structure annotation of LC-MS^2 data. We present a novel machine learning framework called LC-MS^2Struct, which learns to optimally combine the MS^2 and RO information for the accurate annotation of a sequence of MS features. LC-MS^2Struct 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. This way, LC-MS^2Struct learns to optimally use the RO information from a set of LC-MS^2 experiment. We trained and evaluated LC-MS^2Struct on all available reversed phase LC data from MassBank (MB) [37], including a combined total of 4327 molecules from 18 different LC configurations, hence reaching unprecedented measurement diversity in the model evaluation. Our framework is compared with three other approaches: RT filtering, $\log P$ predictions [14], and RO predictions [34]. LC-MS^2Struct can be combined with any MS^2 scorer, and is demonstrated with CFM-ID [12, 10], MetFrag [14, 7] and SIRIUS [11, 9]. The use of chirality encoding circular molecular fingerprints [38] in the predictive model allows to distinguish and rank different stereoisomers based on the observed ROs.

Overview of LC-MS^{2}Struct

Input and output. We consider a typical data setting 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 [39] or PubChemLite [40]) based on the ion's mass, a suspect list, or an in silico molecule generator (e.g. SmiLib v2.0 [41, 42]). 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 [12, 10], MetFrag [14] or SIRIUS [11, 9]. 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 (Figure 1**d**) expressing the agreement between the observed and the predicted RO, for each candidate assignment pair (y_σ, y_τ) given the observed RTs (t_σ, t_τ). 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-marginal scores [43] for each candidate and MS feature, defined for a candidate $m \in \mathcal{C}_\sigma$ and MS feature σ 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]. However, for general graphs G, the max-marginal inference problem (MMAP) is intractable. 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 [43, 44]. 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 [45, 46, 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, 47, 46]. 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

Extracting training data from MassBank. Ground truth annotated MS^2 spectra and RTs were extracted from MassBank [37], a public online database for MS^2 data. Each individual MassBank record typically provides a rich set of meta information (see Supplementary Table 1), such as the chromatographic and MS conditions as well as molecular structure annotations. For training the SSVM model of LC-MS^{2}Struct, the MassBank data was processed such that the experimental conditions were consistent *within* each MS feature set, *i.e.* with identical LC setup and MS configuration as in a typical LC-MS^2 experiment, to ensure comparable RTs, ROs and MS^2 data. We developed a Python package "massbank2db" that can process MassBank records and groups them into consistent MS feature sets, which we denote as MB-subsets. For our experiments, we sampled sequences of MS features from the MB-subsets in order to simulate real LC-MS^2 experiments where the signals of multiple unknown compounds are measured 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.

Comparison of LC-MS^{2}Struct with other approaches. In the first experiment 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. [14], 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. The RO based methods (LC-MS^{2}Struct and MS^2+RO) were trained using the RTs from all available MB-subsets, ensuring that no test molecular structure (based on InChIKey first block) was used for the model training (structure disjoint). For the RT based approaches (MS^2+RT and MS^2+log$P$), the respective predictors were trained in a structure disjoint fashion using only the RT data available for that MB-subset. For the experiment, all MB-subsets with more than 75 (MS^2, RT)-tuples from the ALLDATA data setup were used (see Supplementary Table 2), as the RT based approaches require LC system-specific RT training data. The ranking performance was computed for each LC-MS^2 experiment within a particular MB-subset. The candidate molecules are identified by their InChIKey first block (*i.e.* the structural skeleton), hence no stereoisomers are in the candidate sets.

Each candidate ranking approach was evaluated with three MS^2 scorers: CFM-ID 4.0 [10], MetFrag [14] and SIRIUS [9]. For LC-MS^{2}Struct we use stereochemistry aware molecular fingerprints (3D) to represent the candidates.

Figure 2**a** shows the average ranking performance (top-k accuracy) across 350 LC-MS^2 experiments, each encompassing about 50 (MS^2, RT)-tuples (see Methods). LC-MS^{2}Struct is the best performing method combined with any of the three MS^2 scorers. For CFM-ID and MetFrag, LC-MS^{2}Struct provides 4.7 and 7.3 percentage unit increases over the Only-MS^2 for the top-1 accuracy, corresponding to 80.8% and 106% performance gain. In our setting, that translates to 2.4 and 3.7 additional annotations at the top rank, respectively (out of approx. 50). The performance improvement increases for larger k, reaching as far as 9.3 and 11.3 percentage units at top-20, which means 4.7 and 5.7 additional correct structures, respectively, in the top-20. For SIRIUS,

the improvements are more modest, on average around 2 percentage units for top-1 to top-20. That might be explained by the higher baseline performance of SIRIUS. Nevertheless, SIRIUS can be improved for particular MB-subsets (see Figure 2**b** and the discussion in the next section).

The runner-up score integration method is MS^2+RO, which also makes use of predicted ROs. For CFM-ID and MetFrag it leads to about one-third to a half of the performance gain as LC-MS^2Struct. The approaches relying on RTs, either by candidate filtering (MS^2+RT) or through log$P$ prediction (MS^2+log$P$), lead to only minor improvements for MetFrag and CFM-ID, but none for SIRIUS, for which MS^2+RT even leads 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 with differing LC-MS^2 setups (see Supplementary Table 3) and compound class compositions (see Extended Data Figure 1). We illustrate these differences in Figure 2**b**. Applying LC-MS^2Struct improves the ranking performance in almost all MB-subsets, including the SIRIUS MS^2-scorer (some very slight decreases were observed in some SIRIUS scored 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 seen already in the average results (Figure 2**a**), the benefit of LC-MS^2Struct 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 improvement for CFM-ID. Both datasets are natural product toxins, which are perhaps poorly explained by the bond-disconnection approach of MetFrag. In contrast, for "RP_001" and "UF_003" the largest improvements (top-1) can be reached for CFM-ID. The RT filtering approach (MS^2+RT) performs particularly well for "LQB_000" and "UT_000". These subsets mostly contain lipids and lipid-like molecules (see Extended Data Figure 1).

Since the RT prediction models are trained using only data from the respective MB-subsets, more accurate models may be reached for less heterogeneous subsets of molecules. Hence, the RT filtering could work well in such cases [26].

Performance analysis of LC-MS^2Struct for different compound classifications. Next we investigate how LC-MS^2Struct can improve the annotation across different categories in two molecule classification systems, ClassyFire [51] and PubChemLite [40]. Figure 3 shows the average top-1 and top-20 accuracy improvement of LC-MS^2Struct over the Only-MS^2 baseline for each ClassyFire super-class and PubChemLite annotation category. For ClassyFire (Figure 3**a**), 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 10.8 percentage units for MetFrag, but improves much less for CFM-ID and SIRIUS (1.9 and 3.5 percentage units, respectively). For "Organic oxygen compounds", in contrast, the top-1 accuracy improves by about 10 percentage units when using both CFM-ID and MetFrag, whereas only half that improvement is observed for SIRIUS. This suggests that the CFM-ID results may be improved with the inclusion of more "Alkaloids and derivatives". Additionally, the "Alkaloids and derivatives", "Organic acids and derivatives" and "Organic nitrogen compounds" appear 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 SIRIUS, "Lipids and lipid-like molecules" as well as "Organic oxygen compounds" benefit the most from LC-MS^2Struct in top-1 (both improving by 5.7 percentage units) and top-20 (4.1 and 3.2 percentage units, respectively). In general, for "Lipid and lipid-like molecules", LC-MS^2Struct seems to achieve the best improvement (top-1 and top-20) over all MS^2 scorers. However, depending on the MS^2 scorer this improvement distributes differently across the lipid sub-classes (see Extended Data Figure 2), such as "Fatty Acyls", "Prenol lipids" or "Sphingolipids".

For the PubChemLite classification (Figure 3**b**) we also see that the MS^2 scorers benefit differently from LC-MS^2Struct. The improvement is generally close to the average improvement of the respective MS^2 scorers and seems more equally distributed across the annotation categories.

For example for CFM-ID, the biggest top-1 improvements are in the "Food Related" and

"noClassification" categories. On the other hand, for SIRIUS the "PharmacoInfo" and "BioPathway" categories improve the most. MetFrag shows the most consistent performance improvement across the categories. "AgroChemInfo" benefits the least from LC-MS^{2}Struct (top-1 and top-20) A possible explanation could be that the molecules categorized as agrochemicals are mainly "Benzenoids" (48.5%), "Organoheterocyclic compounds" (25.9%) and "Organic acids and derivatives" (11.6%). As shown in Figure 3**a**, these three ClassyFire classes show low (CFM-ID and MetFrag) or practically no (SIRIUS) improvement when using ROs.

Annotation of stereoisomers. Finally, we study whether LC-MS^{2}Struct can annotate stereoisomers more accurately than MS^2 alone, considering differences between stereoisomers that vary in their double-bond orientation (*e.g. cis-trans* or *E-Z* isomerism), which may potentially lead to differences in their LC-behavior (see Figure 5**a**). 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" (ONLYSTEREO data setup, see Methods). The molecular candidates are represented using two different molecular fingerprints: One that includes stereochemistry information (3D); and one that omits it (2D) (see Methods). This allows us to assess the importance of stereochemistry-aware features for the structure annotation.

Figure 4**a** shows the ranking performance of LC-MS^{2}Struct using 2D and 3D fingerprints.

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%.

Generally, LC-MS^{2}Struct improves the ranking for all three MS^2 scorers. The improvement, however, is notably larger when using stereochemistry-aware (3D) candidate features. Interestingly, a similar behavior could be observed in the ALLDATA setting (see Extended Data Figure 3), even though the absolute performance improvements were smaller. This experiment demonstrates that LC-MS^{2}Struct can use RO information to improve the annotation of stereoisomers.

Discussion

LC-MS^{2}Struct is a novel approach for the integration of tandem mass spectrometry (MS^2) and liquid chromatography (LC) data for the structural annotation of small molecules. The method learns from the pairwise dependencies in the retention order (RO) of MS features within similar LC configurations and can generalize across different, heterogeneous LC configurations. Furthermore, the use of stereochemistry-aware molecular fingerprints enables LC-MS^{2}Struct to annotate stereoisomers in LC-MS^2 experiments based on the observed ROs. Also, our novel processing pipeline to group all (MS^2, RT)-data from MassBank into subsets of homogeneous LC-MS^2 conditions, which is implemented and made available in the "massbank2db" [52] Python package will, we believe, make MassBank more accessible to other researchers and hence lower the bar of entry to computational metabolomics research.

Our experiments demonstrate that LC-MS^{2}Struct annotates small molecules with an accuracy far superior to more traditional retention time (RT) filtering and log$P$-based approaches, and also markedly better than previous methods that rely on ROs. In particular, compared to Bach et al. [34], who used a graphical model as a post-hoc integration tool of MS^2 scores and RO predictions, the benefits of learning the parameters of the graphical model are clear. All three studied MS^2 scorers could be improved by LC-MS^{2}Struct, including the best-of-class SIRIUS, for which improvements have generally been hard to come by due to its already high baseline accuracy. Our results show the superiority of stereochemistry-aware molecular features for the structure annotation of LC-MS^2 data. Remarkably, this was not only the case for the annotation of stereoisomers but also for candidates only distinguished by their 2D structure. This result could be relevant for improving structural annotations in ion mobility separation-mass spectrometry (IMS-MS) with collision cross section (CCS) measurements.

Our examples indicated that LC-MS^{2}Struct separates candidates with varying double-bond stereochemistry, *i.e. E/Z-* and *cis/trans*-isomers (see *e.g.* Figure 5). However, 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 quantify this effect, or interrogate these further until more such data is publicly available. Furthermore, as non-chiral LC cannot distinguish stereoisomers that only differ in their chiral centers, the development of more selective stereochemistry-aware molecular features, ignoring the chiral annotations, might be beneficial. We also note that the direct modelling of a node score (MS^2 information) predictor in the Structured Support Vector Machine would be possible. 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.

Methods

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

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

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 subset 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, 47, 46], 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 $\theta : \mathcal{X} \times \mathcal{Y} \to (0, 1]$ 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 $\Gamma : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathcal{Y} \times \mathcal{Y} \to \mathcal{F}$ 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 [10], MetFrag [14] or SIRIUS [9]. 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 $\phi : \mathcal{Y} \to \mathcal{F}_\mathcal{Y}$ 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 [44].

Feasible inference using random spanning trees (RST). For general graphs G, the maximum a posterior (MAP) inference problem (*i.e.* finding the highest scoring label sequence $\mathbf{y}$ given an MS feature sequence) is an $\mathcal{NP}$-hard problem [53, 54]. The max-marginals inference (MMAP), needed for the candidate ranking, is an even harder problem which is $\mathcal{NP}^{\mathrm{PP}}$ complete [54]. 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 [43, 44]. Such tree-based approximations have shown to be successful in various practical applications [45, 46, 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 influence of the number of SSVM models on the prediction performance is shown in Extended Data Figure 4.

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 [55]:

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

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, 47], 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 [56] following the recent work by Lacoste-Julien et al. [55]. 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:

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

and satisfies $\ell(\mathbf{y}, \mathbf{y}) = 0$ (a required property [55]), if λ is a normalized kernel, which holds true in our experiments (we used the MinMax kernel [57]).

Pre-processing pipeline for raw MassBank records. Extended Data Figure 5 illustrates our MassBank (MB) pre-processing pipeline implemented in the Python package "massbank2db" [52]. 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 were measured under the same LC- and MS-conditions.

Supplementary Table 1 summarizes the grouping criteria. In the next step, we used the InChIKey [58] identifier in MassBank to retrieve the SMILES [59] representation from PubChem [20] (1 Feb., 2021), rather than using the contributor-supplied SMILES. This ensures a consistent SMILES source for the molecular candidates and ground truth annotations.

Three more filtering steps were performed before creating the final database, to remove records: (1) if the ground truth exact mass deviated too far ($>$ 20ppm) from the calculated exact mass based on the precursor mass-per-charge and adduct type; (2) if the subset contained $<$ 50 unique molecular structures; (3) if they were potential isobars (see pull-request #152 in the MassBank GitHub repository, `https://github.com/MassBank/MassBank-data/pull/152`).

Supplementary Table 3 summarizes the LC-MS^2 meta-information for all generated MB-subsets.

Generating the molecular candidate sets. We used SIRIUS [11, 9] 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^{2}Struct to annotate the stereochemical variant of the molecules, we enriched the SIRIUS candidates sets with stereoisomers, using the InChIKey first block of each candidate to search PubChem (1 Feb., 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. [14] or we provided the MS^2 spectra separately (SIRIUS). For the spectra merging we used the "mzClust_hclust" function of the xcms package [60], which first combines all MS^2 spectra's peaks into a single peaklist and subsequently merges peaks based on a mass error threshold.

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. [10]. 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 [61] implemented in the matchms [62] (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 (which is absent in the 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), the matching scores of the candidates associated with a particular MassBank record using in evaluation were predicted using models that did include its ground truth structure (determined by InChIKey first block).

If a MS^2 scorer failed on a MassBank record, we assigned a constant MS^2 score to each candidate.

Molecular feature representations. Extended connectivity fingerprints with function-classes (FCFP) [38] were used to represent molecular structures in our experiments. We employed RDKit (v2021.03.1) to generate *counting* FCFP fingerprints. The fingerprints were computed based on the isomeric SMILES, using the parameter "useChirality" to generate fingerprints that either encoded stereochemistry (3D) or not (2D). 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 3D FCFP fingerprints in our experiments, except for the experiments focusing on the annotation of stereoisomers, where we used both 2D and 3D fingerprints for comparison. We used the MinMax-kernel [57] 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 [40], which classifies molecules according to information available for 10 exposomics-relevant categories. For ClassyFire, we used the "classyfireR" R package to retrieve the classification for each ground truth molecular structure in our dataset.

For PubChemLite, the classification categories were retrieved via InChIKey first block matching of each molecular structure; if it was not found in PubChemLite, the category "noClassification" was assigned.

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 [63], 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 Supplementary Table 2.

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 Supplementary Table 2) as follows:

$$n = \begin{cases} 0 & \text{if } |\mathcal{D}| < 30 \\ 1 & \text{else if } 30 \leq |\mathcal{D}| \leq 75 \\ 15 & \text{else if } 76 \leq |\mathcal{D}| \leq 250 \\ \left\lfloor \frac{|\mathcal{D}|}{50} \right\rfloor & \text{else.} \end{cases}$$

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 Supplementary Table 2).

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 to (maximum) 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. We ensure that the correct candidate is included in the sub-sample. 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) [64] for each C value. The regularization parameter with the highest NDCG value was chosen to train the final model. We used the scikit-learn [65] (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 [11]: 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^{2}Struct, etc.) for each molecular structure in the candidate set (identified by PubChem CID). Depending on the data setup (see Supplementary Table 4) we first collapse the candidates by InChIKey first block (ALLDATA, method comparison and molecule category analysis) or full InChIKey (ONLYSTEREO stereochemistry prediction), assigning the maximum candidate score for each InChIKey first block or InChIKey group, respectively. Subsequently, we compute the top-k accuracy based on the collapsed candidate sets.

For the performance analysis of individual molecule categories, either ClassyFire [51] or PubChemLite [40] 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^{2}Struct 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^{2}Struct with other approaches. We compared LC-MS^{2}Struct 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 (RO) 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 [65] (v0.24.1) implementation of the Support Vector Regression (SVR) [66] 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. [14] 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. [14], 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, 67] (v0.5.0). We used counting `substructure` fingerprints calculated using CDK (v2.5) [68] and the MinMax kernel [57]. 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 [69]. 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 [70] 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 Supplementary Table 2). 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 [71] (`https://zenodo.org/record/5854661`). The candidate rankings of all LC-MS^2 experiments are available online: ALLDATA [72] (`https://zenodo.org/record/6451016`) and ONLYSTEREO [73] (`https://zenodo.org/record/6037629`).

Code availability

The source code developed for this study is available on GitHub, including the implementation of LC-MS^2Struct [74] (v2.13.0, `https://github.com/aalto-ics-kepaco/msms_rt_ssvm`); scripts to run the experiments [75] (`https://github.com/aalto-ics-kepaco/lcms2struct_exp`); and the library implementing the MassBank pre-processing [52] (v0.9.0, `https://github.com/bachi55/massbank2db`). The candidate fingerprints were computed by the ROSVM Python library [67] (v0.5.0, `https://github.com/bachi55/rosvm`) using RDKit (2021.03.1). 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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[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] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

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

[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] Eric Bach. *massbank2db: Build a machine learning ready SQLite database from MassBank.* Version 0.9.0. Jan. 2022. DOI: 10.5281/zenodo.7029738. URL: https://github.com/bachi55/massbank2db.

[53] 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.

[54] 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.

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

[56] 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.

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

[58] 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.

[59] 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.

[60] H. P. Benton et al. "XCMS2: Processing Tandem Mass Spectrometry Data for Metabolite Identification and Structural Characterization". In: *Anal. Chem.* 80.16 (Aug. 2008), pp. 6382–6389. ISSN: 0003-2700. DOI: 10.1021/ac800795f. URL: https://doi.org/10.1021/ac800795f.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

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

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

[67] Eric Bach. *Retention Order Support Vector Machine (ROSVM)*. Version 0.5.0. Aug. 2022. DOI: 10.5281/zenodo.7029744. URL: https://github.com/bachi55/rosvm.

[68] 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.

[69] 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.

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

[71] Eric Bach. *Dataset: "Joint structural annotation of small molecules using liquid chromatography retention order and tandem mass spectrometry data"*. Zenodo, Jan. 2022. DOI: 10.5281/zenodo.5854661. URL: https://doi.org/10.5281/zenodo.5854661.

[72] Eric Bach. *Result files (ALLDATA): "Joint structural annotation of small molecules using liquid chromatography retention order and tandem mass spectrometry data with LC-MS²Struct"*. Zenodo, Apr. 2022. DOI: 10.5281/zenodo.6451016. URL: https://doi.org/10.5281/zenodo.6451016.

[73] Eric Bach. *Result files (ONLYSTEREO): "Joint structural annotation of small molecules using liquid chromatography retention order and tandem mass spectrometry data"*. Zenodo, Feb. 2022. DOI: 10.5281/zenodo.6037629. URL: https://doi.org/10.5281/zenodo.6037629.

[74] Eric Bach. *msms_rt_ssvm: Implementation of the LC-MS²Struct algorithm.* Version 2.13.0. Aug. 2022. DOI: 10.5281/zenodo.7029714. URL: https://github.com/aalto-ics-kepaco/msms_rt_ssvm.

[75] Eric Bach. *Experiments and figure generation for the LC-MS²Struct evaluation.* Sept. 2022. DOI: 10.5281/zenodo.7057255. URL: https://github.com/aalto-ics-kepaco/lcms2struct_exp.

Acknowledgments

The work by E.B. and J.R. was partially 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 MS² 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.

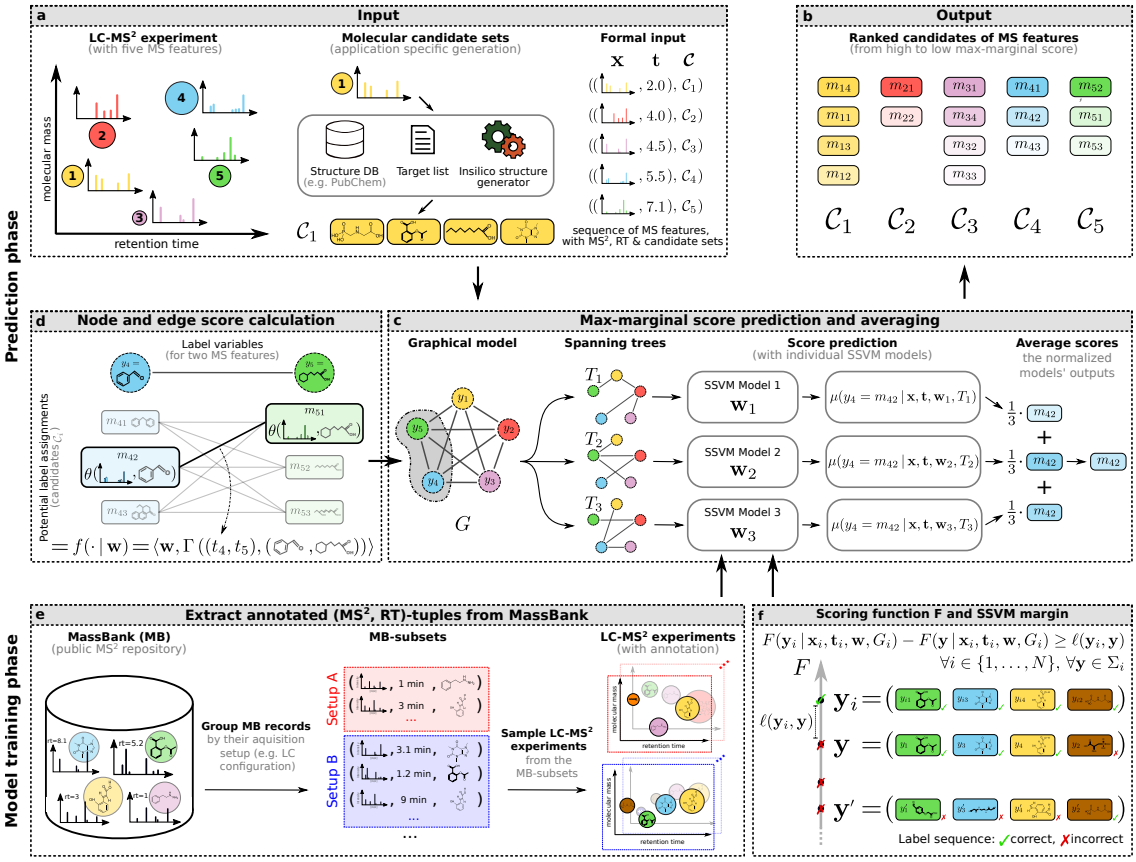


Figure 1: **Overview of the LC-MS²Struct workflow.** **a**: Input to LC-MS²Struct during the application phase. The LC-MS² experiment results in a set of (MS², RT)-tuples. The MS information is used to generate a molecular candidate set for each MS feature. **b**: Output of LC-MS²Struct are the ranked 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 score 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, simulating LC-MS² experiments. **f**: Main objective optimized during the SSVM training, where $\mathbf{y}_i \in \Sigma_i$ is the ground truth label sequence of example i and $\mathbf{y}, \mathbf{y}' \in \Sigma_i$ are further possible label sequences.

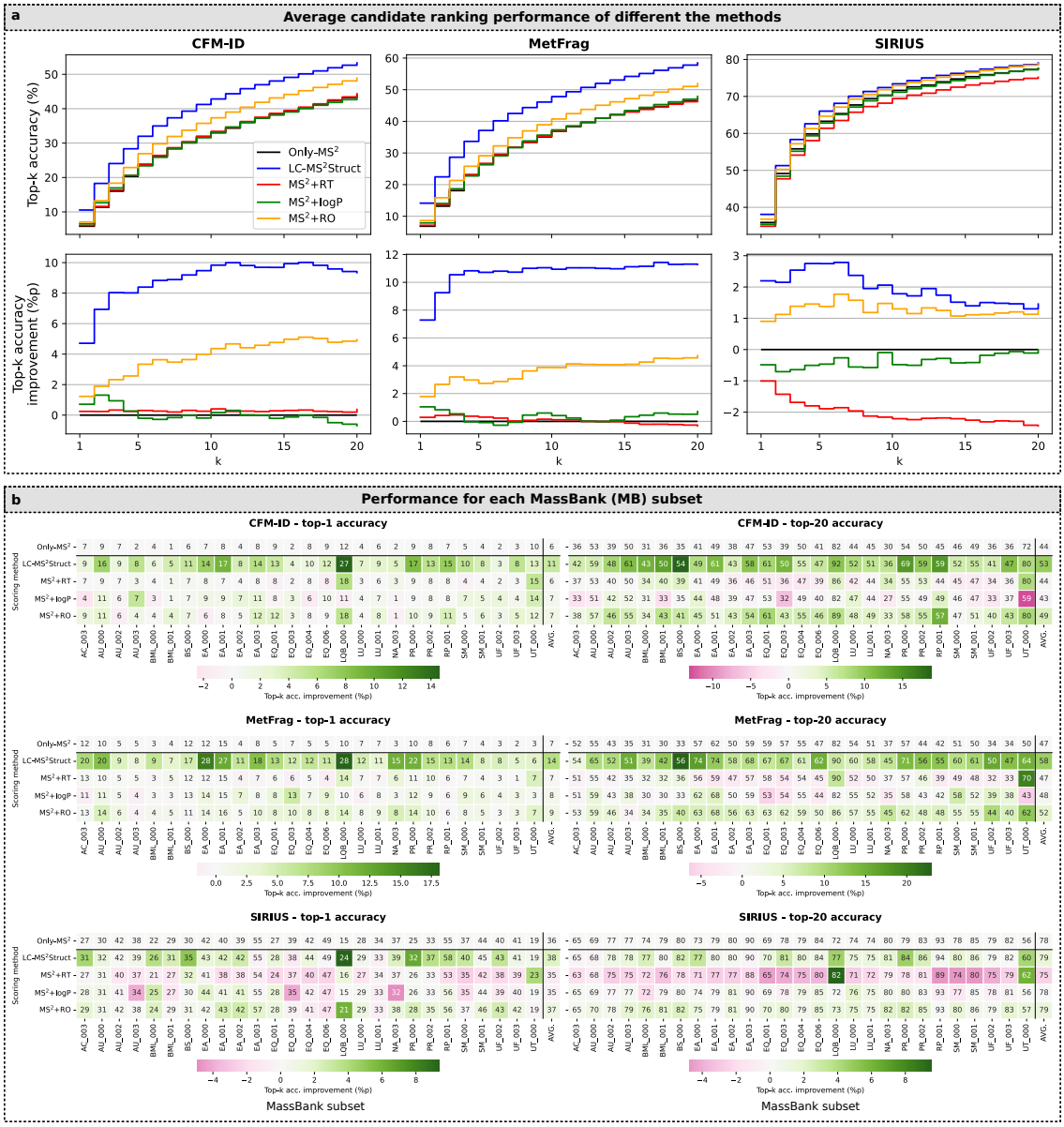


Figure 2: **Different approaches to combine MS^2 and retention time (RT) information: a**: Comparison of the performance, measured by top-k accuracy, for the different ranking approaches combining MS^2 and RT information. The results shown are averaged accuracies over 350 sample MS feature sequences (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 with Only-MS^2.

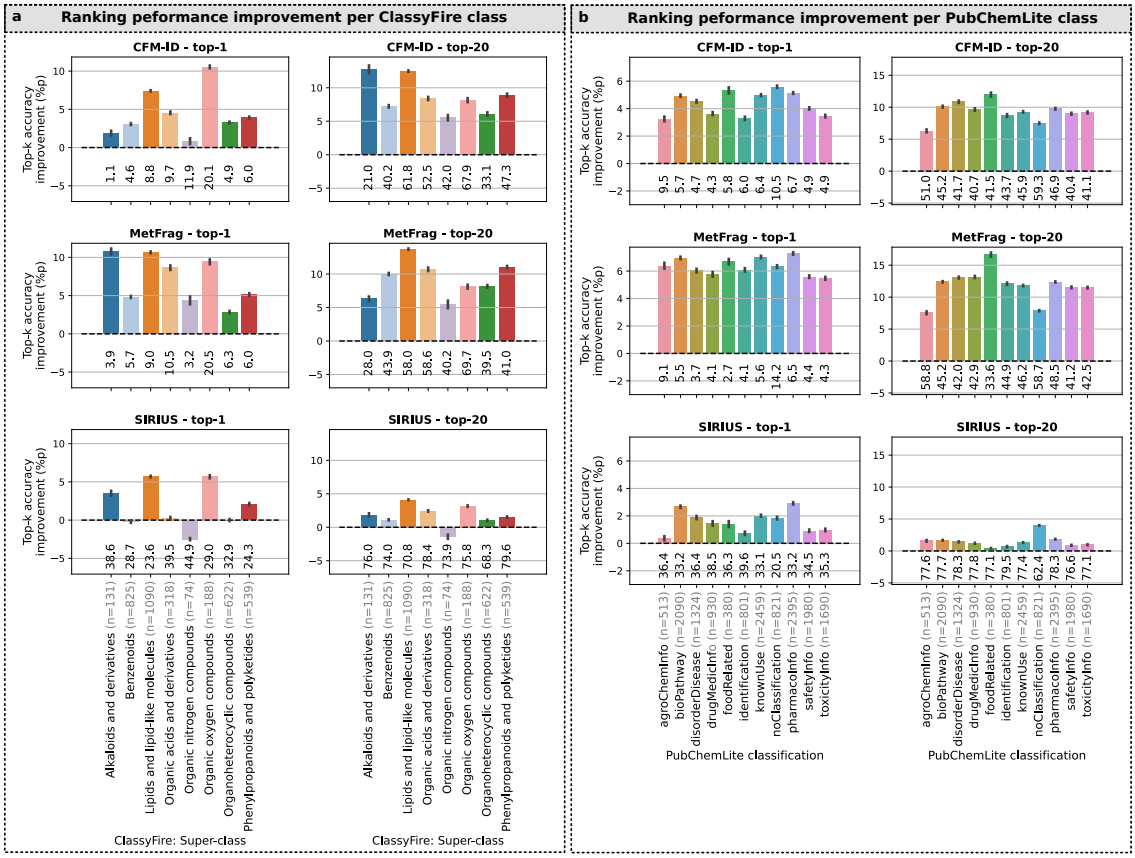


Figure 3: **Performance gain by LC-MS^{2}Struct across molecular classes.** The figure shows the average (50 samples) and 95%-confidence interval (1000 bootstrapping samples) of the ranking performance (top-k) improvement of LC-MS^{2}Struct compared to Only-MS^2 (baseline). The top-k accuracies (%) under 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 (Class level). **b**: PubChemLite [40] 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.

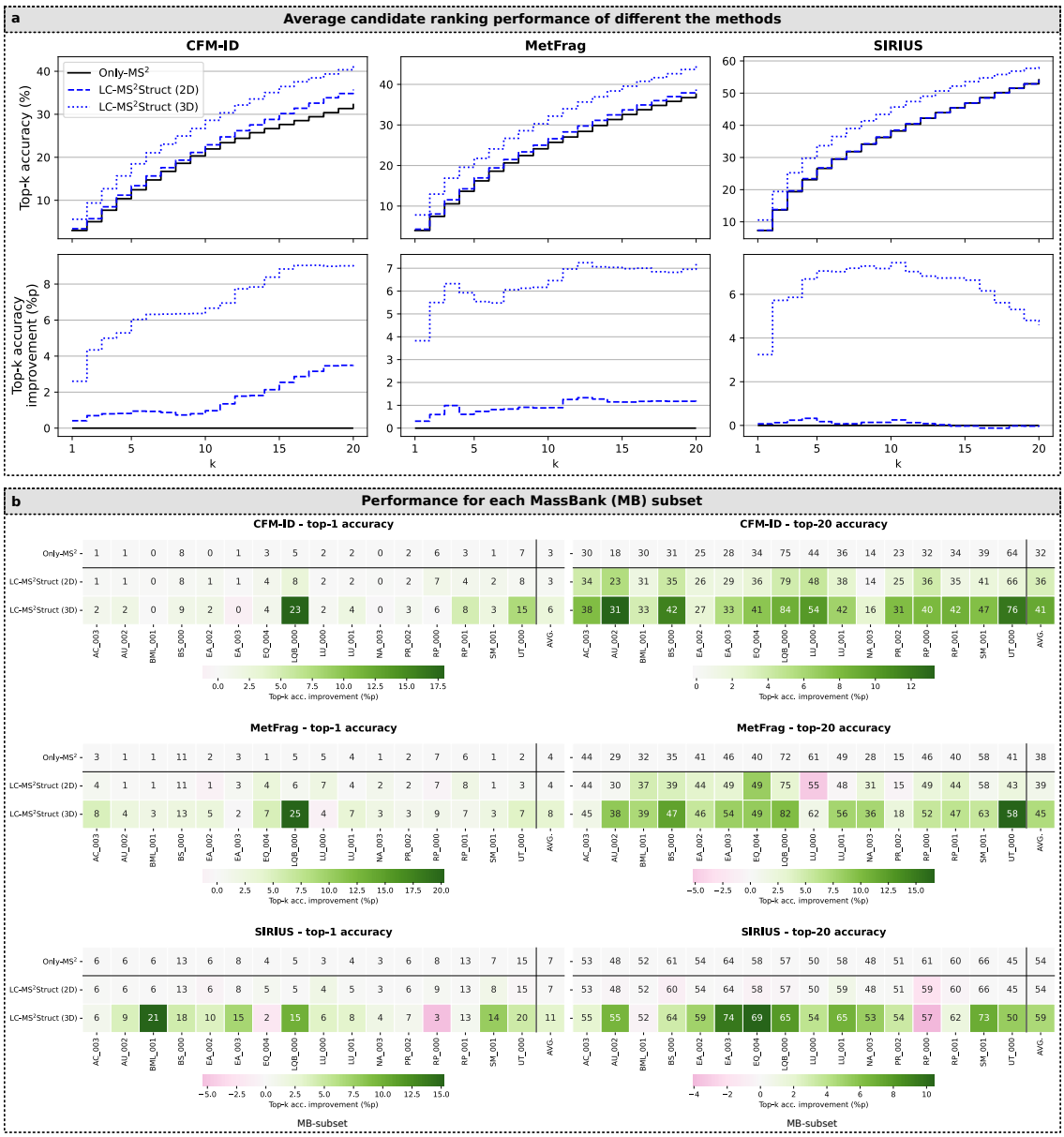


Figure 4: **Using LC-MS^{2}Struct to identify stereoisomers.** **a**: Comparison of the performance, measured by top-k accuracy, of LC-MS^{2}Struct using either 2D (no stereochemistry) or 3D (with stereochemistry) molecular fingerprints in the ONLYSTEREO setting. The results shown are averaged accuracies 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 with Only-MS^2.

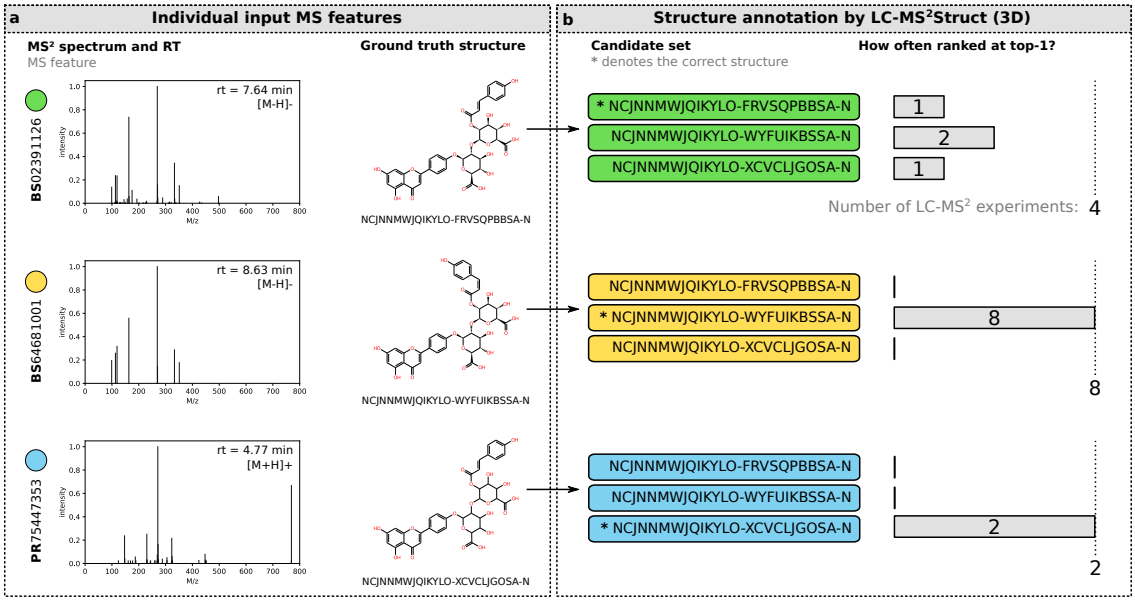
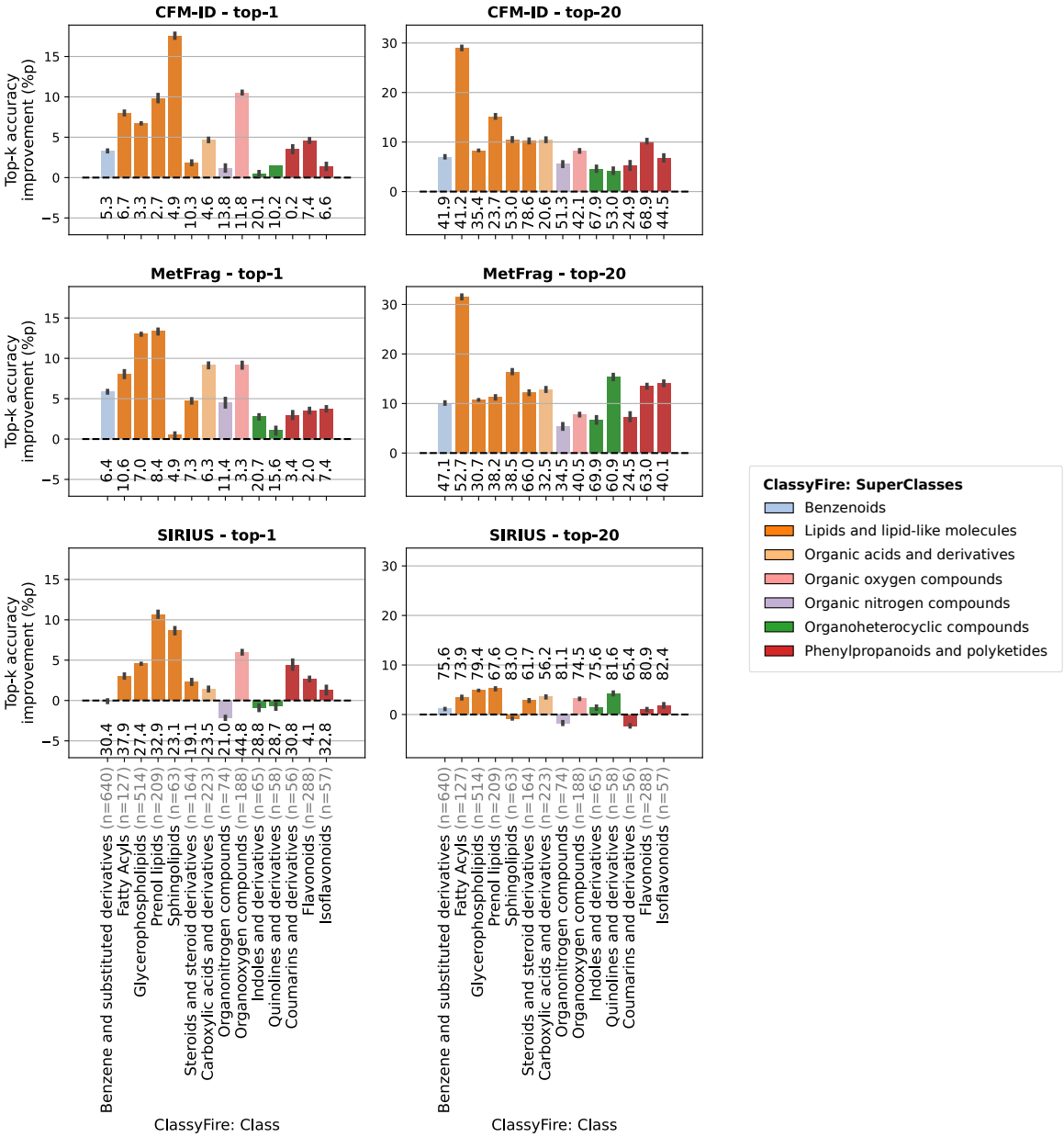


Figure 5: **Application of LC-MS^{2}Struct to annotate stereoisomers.** Post-hoc analysis of the stereoisomer annotation using LC-MS^{2}Struct for three (MS^2, RT)-tuples from our MassBank data associated with the same 2D skeleton (InChIKey first block). In our evaluation, all three MS features were 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.

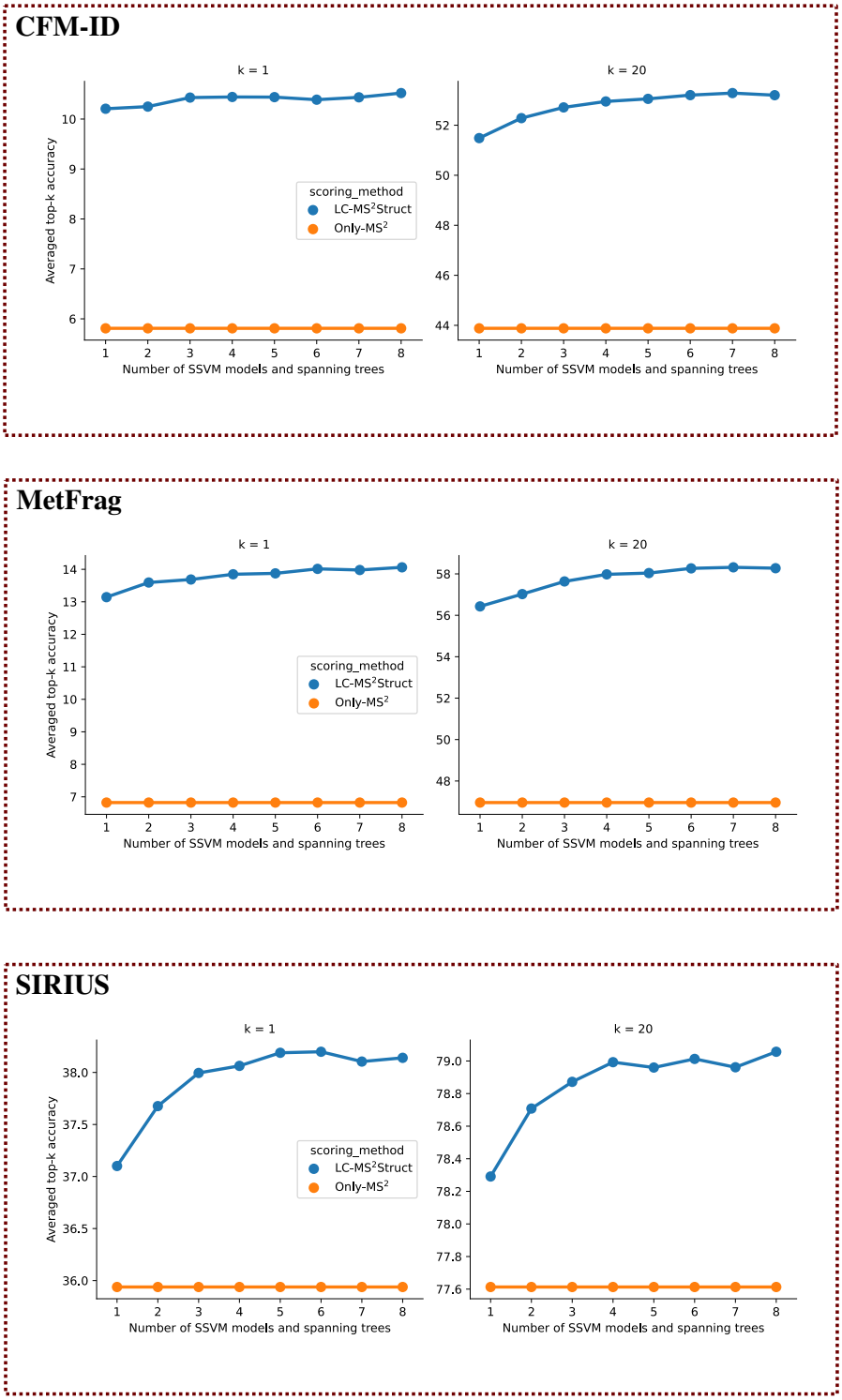
Extended data figures

Extended Data Figure 1: Distribution of molecule classes in the MassBank (MB) subsets. ClassyFire SuperClass distribution [51] for each MB-subset studied in our experiments. Within each MB-subset, the label "Other" is assigned to each SuperClass which makes up less then 2.5% of all molecules. The center label represents the number of examples for the respective MB-subset.

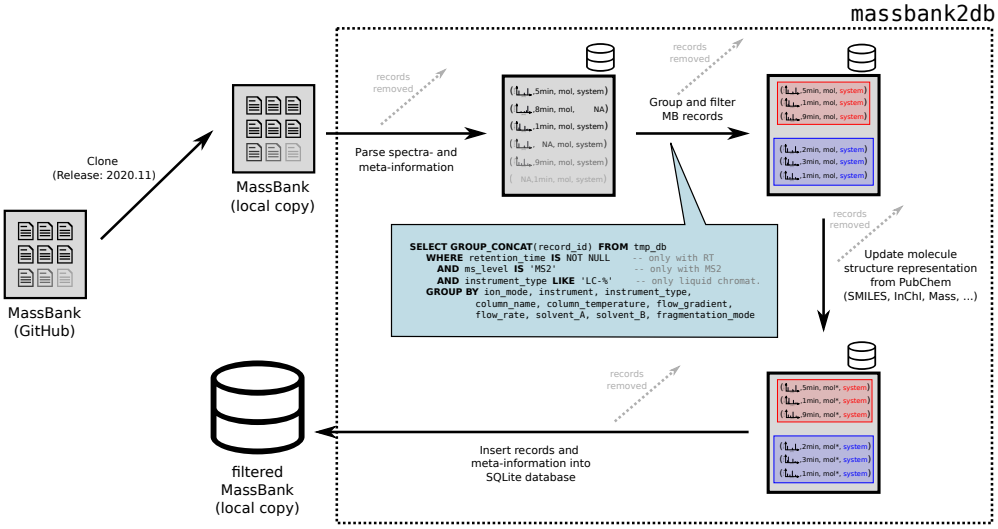


Extended Data Figure 2: Performance gain by LC-MS^{2}Struct across ClassyFire Class-level annotations. The Classes (shown in the bars) are colour coded by SuperClasses (see legend). The figure shows the average (50 samples) and 95%-confidence interval (1000 bootstrapping samples) of the ranking performance (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).

Extended Data Figure 3: Using LC-MS^{2}Struct with different molecule feature representations to identify the correct structure at the level of first InChIKey block (InChIKey-1). **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 in the ALLDATA setting. The results shown are averaged accuracies over 354 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.



Extended Data Figure 4: Performance comparison of LC-MS^{2}Struct against using only MS^2 information (Only-MS^2) for different number of SSVM models. The performance curves for the three MS^2-scorers are shown separately. The top-k accuracies shown are averaged accuracies over 354 sample MS feature sequences (LC-MS^2 experiments) from the ALLDATA setting.



Extended Data Figure 5: 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 [52].