# Learning to Fragment Molecular Graphs for Mass Spectrometry Prediction

# *Learning to Fragment Molecular Graphs for Mass Spectrometry Prediction*


A THESIS PRESENTED
BY
JANET LI
TO
THE DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE JOINT DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECTS OF
COMPUTER SCIENCE AND STATISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
MAY 2023

Thesis advisor: Connor Coley Janet Li

# *Learning to Fragment Molecular Graphs for Mass Spectrometry Prediction*

Abstract

Mass spectrometry is a common technique used to identify metabolites, small molecules that are yielding important new insights into a number of important biological and physiological processes, including organ function, nutrient sensing, and gut physiology. However, identifying these metabolites is a difficult task. Most methods for metabolite identification compare the spectrum of an unknown compound against a database containing referential spectra of known compounds. However, this approach fails if the true compound is not in the reference database, motivating approaches to generate a larger standard library by directly learning to simulate the forward fragmentation process. In this thesis, I propose a machine learning framework that learns the fragmentation pathways for each molecule to predict its mass spectrum, drawing upon both advances in machine learning and domain insights from analytical chemistry.

# Contents

# Listing of figures

# Acknowledgments

First and foremost, I would like to thank my advisor Sam Goldman, whose guidance and endless patience were invaluable. This thesis also would not have been possible without support from Professor Connor Coley, who provided me with the opportunity to conduct this research. I would also like to thank Professor Lucas Janson for providing additional guidance and insights. Last, but not least, many thanks to my friends and family for their constant support and encouragement throughout this journey.

# 1 Introduction

Machine learning has transformed computer vision [26] and natural language processing [10], and has also undergone a renaissance in chemistry, with a focus on reaction prediction [3], property prediction [32], and synthesis planning [20]. Analytical chemistry, the study of the composition and structure of matter, is particularly poised to take advantage of advances in deep learning for graphs, as many longstanding analytical chemistry techniques used to identify unknown molecules, such as nuclear magnetic resonance (NMR) [29] and mass spectrometry [9], can be framed as convolutions on molecular graphs.

In this thesis, I focus on mass spectrometry, a standard high-throughput technique for analyzing unknown molecules, with diverse applications including metabolomics [4], drug discovery [13], and forensic and clinical toxicology [15]. A significant bottleneck in mass spectrometry experiments is the interpretation of the resulting spectra to determine the structure of the molecule in question.

Drawing on insights from analytical chemistry, I propose a machine learning framework that first learns how molecules can fragment, then uses this knowledge to predict the mass spectrometry spectrum of a given molecule.

The remaining chapters are structured as follows: Chapter 2 discusses background information about metabolomics and mass spectrometry, and related work; Chapter 3 formalizes the task; Chapter 4 discusses the data labeling process; Chapter 5 discusses the fragmentation model and tree generation process; Chapter 6 discusses the intensity model and resulting predicted spectra; and Chapter 7 provides concluding remarks and directions for future work.

# 2

# Background Information

Metabolomics is a rapidly evolving field of life science that is yielding important new insights into a number of important biological and physiological processes, including organ function, nutrient sensing, and gut physiology [30]. The metabolome is the complete collection of metabolites, or small molecule chemicals, found in a given biological sample (e.g., organelle, cell, organ, biofluid, or organism), and it is inherently large and complex. While the genome consists of combinations of just four nucleotide bases and the proteome consists of combinations of 20 proteogenic amino acids, the metabolome consists of hundreds of thousands of small molecules belonging to thousands of different chemical classes. The human metabolome likely consists of far more than a million compounds, of which only 114,000 have been identified so far [30].

One common technique used to conduct metabolomics experiments and identify new metabolites is liquid chromatography with tandem mass

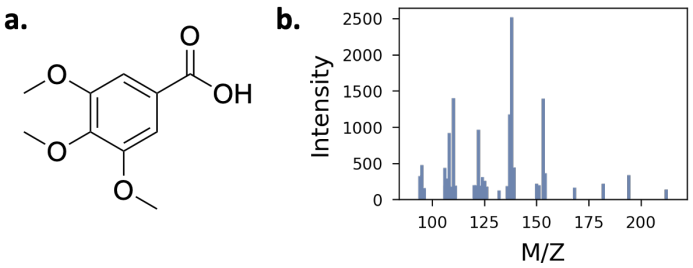


**Figure 2.0.1:** The molecule 3,4,5-trimethoxybenzoic acid **(a)** and its MS/MS spectrum **(b)**.

spectrometry (LC-MS/MS). Liquid chromatography (LC) uses column chromatography to separate different molecules in a complex solution, to then analyze the unknown components individually. In tandem mass spectrometry (MS/MS), the unknown molecule is ionized and its mass is measured (MS1), then introduced into a collision cell, where it fragments from collisions with neutral gas atoms (MS2). The mass-to-charge ratios of the molecular fragments are measured, resulting in a spectrum containing the measured mass to-charged ratios with their corresponding intensities [30]. A molecule and its corresponding mass spectrum are shown in Figure 2.0.1.

A significant bottleneck in mass spectrometry experiments is structural elucidation: the interpretation of the resulting spectrum to determine the chemical structure of the unknown metabolite. Most methods for metabolite identification compare the spectrum of an unknown compound against a database containing referential spectra of known compounds. However, spectral libraries only cover a small fraction of the known chemical space. PubChem, a database of chemical molecules, currently contains information for 112 million unique compounds, whereas all metabolomics spectral libraries combined account for less than 1% of these molecules [2]. As spectral library searching can only annotate known molecules with reference spectra, there are many cases where the spectrum does not match any of the reference spectra in the spectral library and the molecule can not be identified, motivating the development of

new computational techniques. The computational techniques that have been developed for this identification can be broadly categorized into three approaches: clustering, spectrum to molecule, and molecule to spectrum.

Clustering approaches such as Spec2Vec [11] and feature-based molecular networking [19] cluster similar spectra to find groups of closely-related molecules. This results in significant speed-ups when querying the spectrum of an unknown molecule against the spectra in a large database, by only comparing against the representative spectra of the clusters.

Spectrum to molecule approaches such as FingerID [24], CSI:FingerID [6], and MIST [8] directly predict the molecular structure from the spectrum. For example, CSI:FingerID first assigns a chemical formula to each peak in the spectrum and arranges these formulas in a tree structure. This formula tree is used to compute kernels, which are then used to train a support vector machine to predict each molecular property individually, forming a molecular structure fingerprint of the unknown compound. This fingerprint is then queried against a molecular database to identify the specific molecular structure.

On the other hand, molecule to spectrum approaches attempt to predict the mass spectrum from a given molecule. NEIMS [28] treats the spectrum as a fixed length vector of binned intensities, where each intensity represents the sum of observed intensities within a small range. However, such approaches are capable of predicting peaks that don't necessarily correspond to substructures of the true molecule, violating invariants of the problem.

Instead of trying to directly predict a spectrum from a molecule, a more interpretable approach is to instead predict how a molecule will break, and determine which molecule has the closest breakage pattern to a query spectrum. In this thesis, we present one such fragmentation-based approach, so MetFrag [23, 31], MAGMa [21, 22], and CFM-ID [1] are the most relevant existing works to contextualize our contribution. These approaches use in silico fragmentation of molecules to predict their spectra, generating a larger library of labeled metabolite standards. If these prediction tools are accurate, we can then identify metabolites by searching these expanded libraries.

In brief, MetFrag and MAGMa attempt to explain the peaks observed in mass spectrometry spectra by combinatorially generating possible substructures, then matching substructures to spectra peaks by their mass-to-charge ratio and a penalty score. For example, MAGMa determines penalty scores based on the bonds that are broken to form the fragment. However, the combinatorial enumeration of fragmentation possibilities becomes computationally expensive for large input molecules.

CFM-ID [1] proposes a probabilistic generative model, competitive fragmentation modeling (CFM) for the MS/MS fragmentation process. MS/MS fragmentation is modeled as a Markov process, involving state transitions between fragments. CFM is a latent variable model in which the only observed variables are the initial molecule and the output peak; the fragments themselves are never directly observed. To predict a complete mass spectrum, the model is run multiple times to compute the marginal distribution of the output peak. A maximum likelihood approach is used to estimate parameters for CFM from MS/MS data, based on the EM algorithm. However, CFM-ID is extremely expensive to train (one iteration on 60,000 spectra takes 10 hours on a 64-core machine) and is also not that accurate (failing to annotate 42% of monoisotopic peaks in the NIST-20 dataset) [18].

In addition, current molecule to spectra approaches do not attempt to learn the fragmentations themselves: MetFrag and MAGMa combinatorially generate all possible fragments, while CFM-ID treats the fragmentations as latent variables. Thus, a new hybrid approach is necessary that both leverages the advantages of supervised machine learning and predicts full fragmentations, with the goals of improved interpretability, more efficient model training and inference, and higher accuracy.

# 3

# Problem Overview

Formally, mass spectrometry can be represented as a generative process that takes as input some molecule $\mathcal{M}$, and produces a corresponding spectrum $\mathcal{S}$. The peaks in the spectrum can be represented as a list of tuples of mass and intensity $[(m_1, i_1), (m_2, i_2) \ldots, (m_n, i_n)]$, where each observed mass $m_j$ corresponds to a substructure $\mathcal{F}_j$ of the molecule $\mathcal{M}$, and each intensity $i_j$ corresponds to the relative abundance of that substructure. Figure 3.0.1 shows an example annotated spectrum.

In practice, because the inverse function is unknown, comparing an observed spectrum $\mathcal{S}_{\text{observed}}$ to a library of standard $(\mathcal{M}, \mathcal{S})$ pairs is done by finding $\arg\min(D(\mathcal{S}_{\text{observed}}, \mathcal{S}_{\text{standard}}))$ over the standard library for some distance function $D$. However, this approach fails if the true $\mathcal{M}$ is not in the standard library, motivating approaches to generate a larger standard library by directly

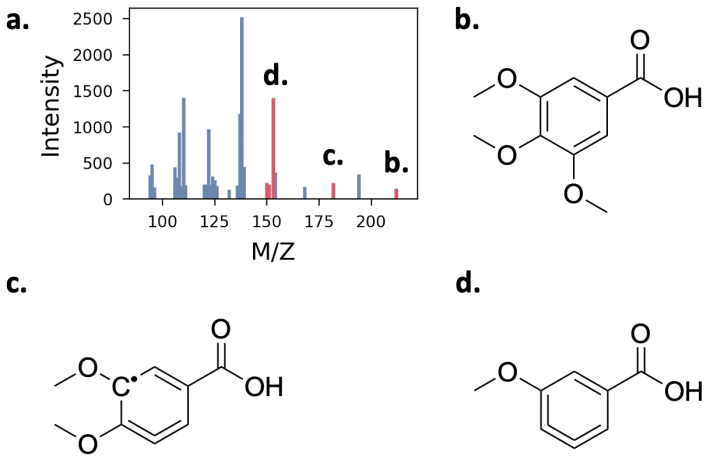


**Figure 3.0.1:** An example annotated spectrum **(a)** for the input molecule 3,4,5-trimethoxybenzoic acid **(b)**. The explained (annotated) peaks are in red, and the unexplained peaks are in blue. The annotated peaks on the left, middle, and right of the spectrum correspond to the structures **(d)**, **(c)**, and **(b)**, respectively.

learning to simulate the forward fragmentation process with a function $f : \mathcal{M} \mapsto \mathcal{S}$. While previous works to date have attempted to solve this problem, their accuracy remains low, motivating new methods development.

At a high level, I pose the problem of predicting spectra as the task of generating molecular fragments related to each other in a tree structure and predicting the intensity value for each fragment. We build a dataset of fragmentation trees and intensity values using data from a metabolite spectral library [27]. We propose to reconstruct such trees using a two-step modeling process: the first step learns the fragmentation pathways for each molecule, and the second step learns to predict the intensities corresponding to each fragment.

## 3.1 Problem Formulation

We start from $\mathcal{M}$, the root molecule, and model the progressive bond breakages in $\mathcal{M}$ to produce a tree of fragments (substructures) of $\mathcal{M}$. An example

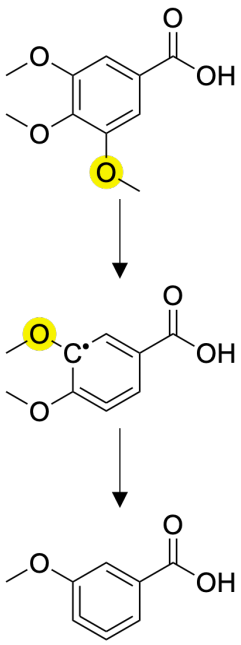


**Figure 3.1.1:** Example fragmentation tree for 3,4,5-trimethoxybenzoic acid. The atom that is removed is highlighted in yellow.

fragmentation tree is shown in Figure 3.1.1.

We define the molecular graph of a candidate molecule $\mathcal{M}$ as an undirected graph $\mathcal{M} = (V, E)$, where $V$ is the set of $N$ vertices representing the $N$ atoms in the molecule, and $E \subset V \times V$ is the set of edges, where $(u, v) \in E$ if there is a bond connecting the atoms corresponding to nodes $u$ and $v$ in the molecule $M$.

To fragment $\mathcal{M}$, we remove each node sequentially to produce $n$ fragment subgraphs $\mathcal{F}_1, \ldots, \mathcal{F}_n$, where $\mathcal{F}_i = (V_i, E_i)$ has nodes $V_i = V \setminus \{v_i\}$ and edges $E_i = E \setminus \{(v_i, u) | u \in V\}$. Figure 3.1.2 shows an example fragmentation step (a single node removal) for a molecule and its corresponding molecular graph, where the removal of a secondary atom (a carbon atom bonded to two other carbon atoms) results in 2 fragment substructures.

In the mass spectrometer, when molecules are fragmented, the hydrogen atoms in the fragments may undergo rearrangements for stability (e.g., McLafferty rearrangements [16]). For example, a hydrogen rearrangement can lead to changes in bond order (e.g., a single bond becoming a double bond) or ring closures, which increase the stability of the substructure, as shown in Figure 3.1.3. These hydrogen rearrangements change the bonds in the molecule, and will modify the edges of the molecular graph. It is difficult to directly include these rearrangements because it is unclear where exactly in the molecule these changes will occur. Instead, we implicitly account for these arrangements by allowing for a deviation in mass corresponding to at most two hydrogen atoms when matching substructures to peaks. The mass of a single hydrogen atom is 1.008 amu (atomic mass units), so a substructure $\mathcal{F}_j$ with true mass $m_j$ can be matched to 5 masses $(m_j - 2.016, m_j - 1.008, m_j, m_j + 1.008, m_j + 2.016)$ and 5 intensity values, corresponding to the 5 hydrogen deviations $\Delta H = -2, -1, 0, 1, 2$. The masses and intensities determine the locations and heights of the peaks in the spectrum, respectively.

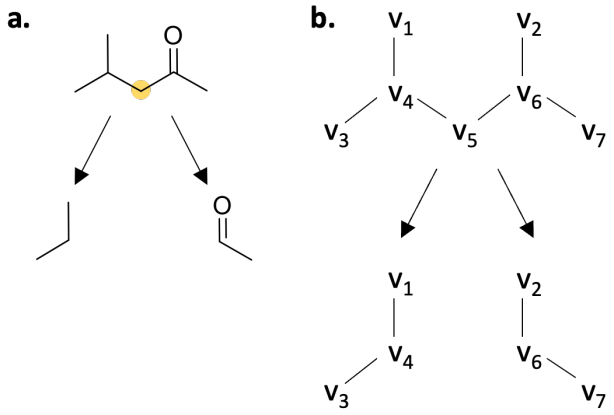


**Figure 3.1.2:** Example fragmentation step **(a)** and the corresponding molecular grpahs **(b)**. A secondary atom (highlighted) is removed to produce two substructures.

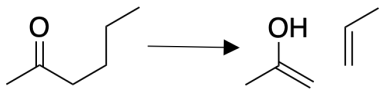


**Figure 3.1.3:** Example of a hydrogen rearrangement (McLafferty rearrangement) resulting in a change in bond order. The expected substructure (left) is not observed, but its rearrangement (right) is.

# 4

# Data Labeling

## 4.1 GNPS Dataset

We empirically evaluate our model on a dataset of 3,863 MS/MS spectra from the public Global Natural Products Social Molecular Networking database (GNPS) [27], as prepared by Duhrkop et al.[6]. The dataset contains natural products (molecules made by living organisms) from the following GNPS libraries, retrieved in December 2014: FDA Library Pt 1 and Pt 2, PhytoChemical Library, NIH Clinical Collections 1 and 2, NIH Natural Products Library, Pharmacologically Active Compounds in the NIH Small Molecule Repository, and Faulkner Legacy Library provided by Sirenas MD. Molecules with mass above 1,010 amu and spectra containing fewer than five peaks with relative intensity above 2% were discarded.

## 4.2 Data Labeling Process

A key component to training neural networks to generate the fragmentations and the intensities is a labeled dataset for supervised learning. Metabolite spectral libraries contain datasets of molecules and their corresponding spectra, which we will use build a dataset of standardized fragmentation trees, as such fragmentations are initially unobserved. Each node of the fragmentation tree represents a molecular substructure (a subgraph of the root molecule, not only a chemical formula), and corresponds to a set of peaks in the resulting spectrum.

First, we combinatorially generate the possible substructures of each candidate molecule, by sequentially removing nodes of the molecular graph $\mathcal{M}$. Consecutive fragmentation steps are performed according to a breadth-first algorithm. We then organize the substructures into a hierarchical fragmentation tree, where each tree node is a substructure, and there is an edge from substructure $\mathcal{F}_i$ to substructure $\mathcal{F}_j$ if $\mathcal{F}_j$ can be obtained by removing a single atom from $\mathcal{F}_i$.

We avoid the exponential time complexity of generating all possible substructures by constraining the edges in the fragment graphs, and by restricting the number of fragmentations. First, the edges in each fragment graph are constrained by the edges in the parent molecule, implicitly assuming that the atoms in the fragment molecule are bonded in the same way as in the parent structure. Second, we restrict fragments to those formed by breaking at most two bonds in the root molecule $\mathcal{M}$ (i.e., each fragmentation tree has a maximum tree depth of 3). Breaking more bonds results in a more complete set of fragments, but also causes the generation of more unlikely fragments.

Now that we have generated the possible substructures, we must determine which substructure corresponds to each peak in the mass spectrum. For each peak $(m_j, i_j)$ in the spectrum, substructures are identified for which the calculated masses match the observed mass $m_j$ within a tolerance of 2.016 amu (corresponding to a deviation of 2 hydrogens), to account for hydrogen rearrangements. We calculate the mass of a molecular graph or fragment graph by

summing the masses of all the atoms corresponding to the nodes in the graph.

After this assignment, each peak in the spectrum will have a set of potential substructures from the fragmentation tree that can explain its presence. In order to select a single substructure for each peak, we use a heuristic to assign a penalty score to each substructure [22]. The substructure penalty score, *SPS*, is calculated by summing the penalty values of all bonds that are disconnected to form the substructure. The penalty value for a disconnected bond $e$ depends on the type of bond that is involved ($p_e$), i.e., whether $e$ is a single, double, triple or aromatic bond, and whether or not $e$ involves a non-carbon atom ($h_e = 2$ if $e$ is a carbon-carbon bond, and $h_e = 1$ otherwise):

$$SPS = \sum_e h_e p_e. \tag{4.1}$$

The substructure with the lowest penalty score is selected as the one that explains the observed peak.

We prune the fragmentation tree by removing sub-trees that do not contain any substructures that explain peaks in the spectrum. To reduce the number of duplicate substructure nodes, we restructure the tree as a directed acyclic graph. Finally, for each substructure in the fragmentation tree, we compute 5 relative intensities (normalized to be maximum 1 for each spectrum), one for each possible hydrogen deviation. Figure 4.2.1 shows an example spectrum and the corresponding fragmentation tree.

## 4.3 Data Labeling Results

First, we clean the spectra by filtering it to the top 60 peaks in each spectrum that can be explained by chemical formulae, using the first part of the preexisting SIRIUS processing pipeline [5]. To do so, it labels fragment peaks with sub-formulas of the target molecule, inferred from their measured mass-to-charge ratios.

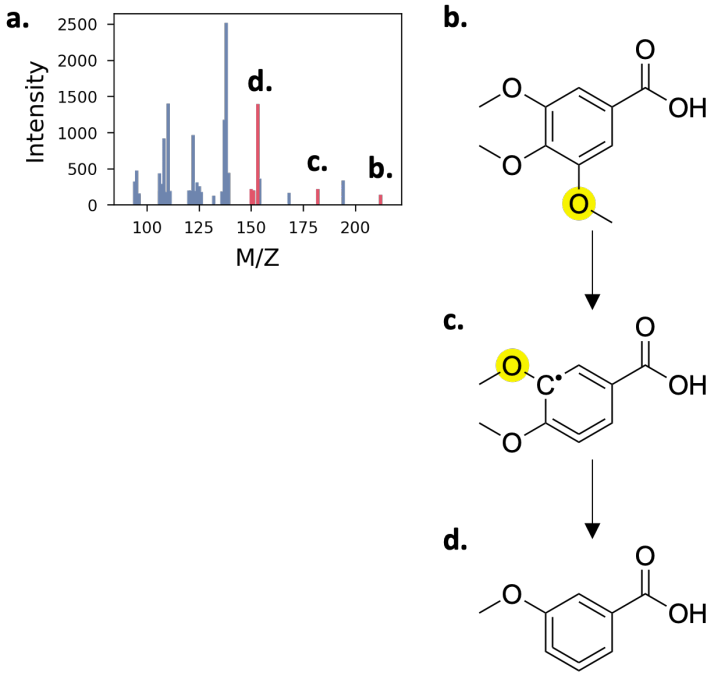


**Figure 4.2.1:** Example spectrum **(a)** and corresponding fragmentation tree **(b-d)** for 3,4,5-trimethoxybenzoic acid **(d)**. In the spectrum, the explained (annotated) peaks are in red, and the unexplained peaks are in blue.

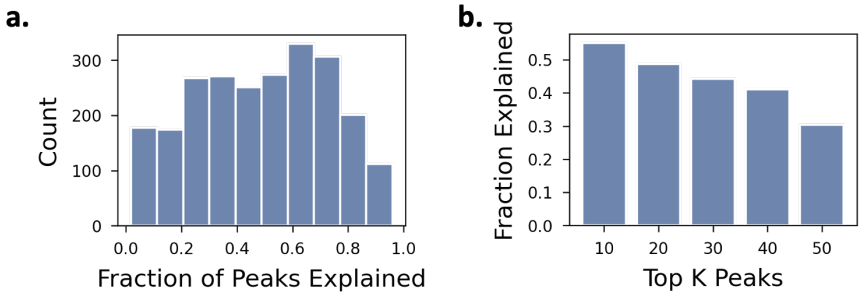


**Figure 4.3.1:** Data labeling pipeline explains peaks in MS/MS spectra. **a.** Distribution of the fraction of peaks explained in each spectrum. **b.** Average fraction of the top $k$ peaks (peaks with the $k$-highest intensities) that are explained, for $k = 10, 20, 30, 40, 50$.

With cleaned spectra in hand, we must establish which molecular substructure corresponds to each subpeak. To do so, we apply our fragmentation tree building algorithm on each molecule-spectrum pair to derive an annotated dataset for supervised learning. On average, we annotate 48.9% of the peaks in each spectrum with substructures, and are able to explain a higher fraction of the top peaks, peaks with larger relative intensities (Figure 4.3.1, 4.3.2). The resulting fragmentation trees are also reasonably sized, with most trees containing fewer than 20 substructures (Figure 4.3.3). Such trees are within the limit of current graph generation capabilities [14].

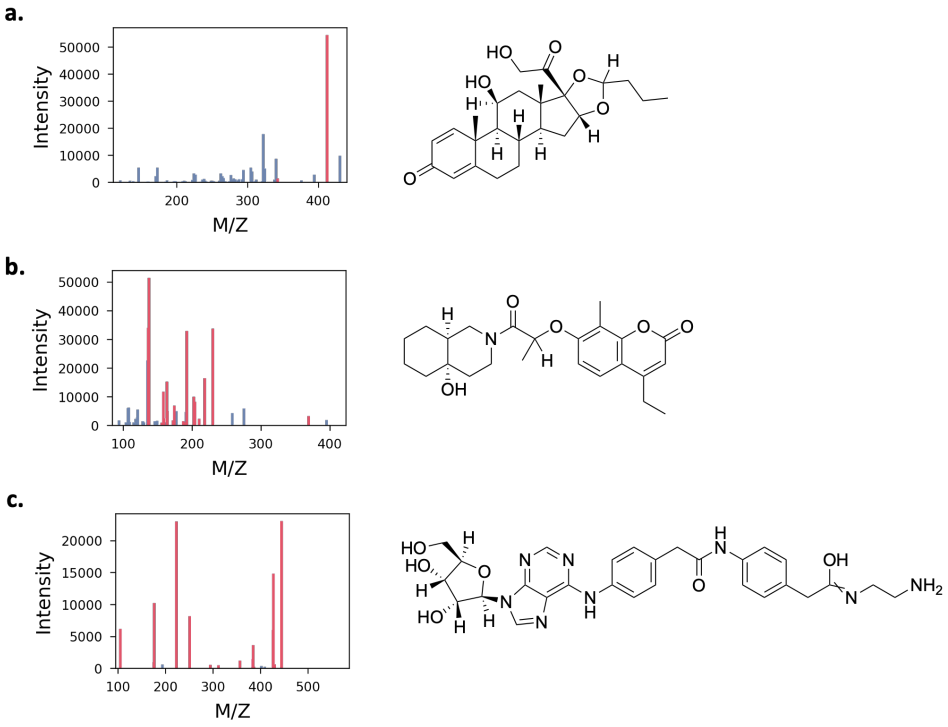


**Figure 4.3.2:** Example mass spectra as annotated our data processing pipeline. Explained peaks are shown in red, with unexplained peaks shown in blue. The data labeling scheme explains 3.5% **(a)**, 46.5% **(b)**, and 78.9% **(c)** of the peaks for each spectrum, respectively. Each input molecule is shown to the right of its corresponding spectrum.

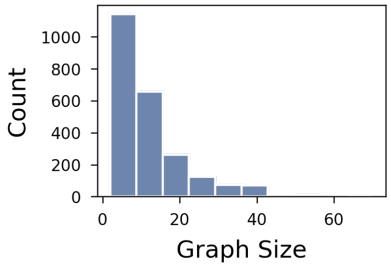


**Figure 4.3.3:** Distribution of the size (number of nodes or fragments) of the annotated fragmentation trees.

# 5

# Fragmentation Tree Generation

The first task in the model pipeline is to predict the fragmentation tree of the input molecule.

## 5.1 Problem Formulation

Because the fragment space is vast, we are unable to completely enumerate all potential fragments $\mathcal{F}$ for a molecule $\mathcal{M}$. Instead, we will sample the set of states conditional upon the root molecule by defining $P(\mathcal{F}|\mathcal{M})$, parameterized by a learned neural network. For some set of fragments $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n$, we have

$$P(\mathcal{F}|x) = P(\mathcal{F}_1|x) \prod_{i=1}^{n} P(\mathcal{F}_{i+1}|\mathcal{F}_i, x). \tag{5.1}$$

We will use the neural network to predict $P(\mathcal{F}_{i+1}|\mathcal{F}_i, \mathcal{M})$, the probability of obtaining fragment $\mathcal{F}_{i+1}$, given the preceding (parent) fragment $\mathcal{F}_i$ and the root molecule $\mathcal{M}$.

## 5.2 Message-Passing Networks

To build a fragmentation tree, a model must learn which atom to remove at every fragmentation step, i.e., which node to remove from the molecular graph. A core component of doing such is to build a representation at each node, using information from nearby nodes. The message-passing network framework [7] has been shown to be effective in learning features from molecular graphs for molecular property prediction, and is therefore selected in this work.

The message-passing network is comprised of a single graph convolution layer that is reapplied for $T$ iterations and is defined in terms of message function $M_t$ and vertex update function $U_t$. During each message passing step, hidden states $h_v^t$ at each node in the graph are updated based on messages $m_v^{t+1}$ according to

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{5.2}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}), \tag{5.3}$$

where $N(v)$ denotes the neighbors of $v$ in the graph.

The initial node (atom) features $a_v^\circ \in \mathbb{R}^{74}$ are concatenations of one-hot encodings of the atom type (element), atom degree, number of implicit hydrogens, formal charge, number of radical electrons, atom hybridization, whether the atom is aromatic, and total number of hydrogens on the atom. We project the initial node features into the hidden node size $d_n$ using the linear transformation $h_v^\circ = W_\circ a_v^\circ$, where $W_\circ \in \mathbb{R}^{d_n \times 74}$. The edge features $e_{vw} \in \mathbb{R}^4$ are one-hot encodings of the bond type between atoms $v$ and $w$: single, double, triple, or aromatic.

The message function is defined $M(h_v, h_w, e_{vw}) = A(e_{vw})h_w$ where $A(e_{vw})$ is a

neural network which maps the edge vector $e_{vw}$ to a $d_n \times d_n$ matrix, where $d_n$ is the dimension of the internal hidden representation of each node in the graph. Specifically, $A$ consists of a linear layer to transform the input edge features from size 4 to size $d_e$, then a non-linear activation function (rectified linear unit, ReLU), then another linear layer to transform the $d_e$ representations at each node to $d_n \times d_n$ representations. Mathematically,

$$A(e_{vw}) = W_e^1 \max[0, W_e^0 e_{vw}], \tag{5.4}$$

where weights $W_e^0 \in \mathbb{R}^{d_e \times 4}$ and $W_e^1 \in \mathbb{R}^{(d_n \times d_n) \times d_e}$, and $A(e_{vw}) \in \mathbb{R}^{d_n \times d_n}$ (after the output is reshaped from $d_n \times d_n \times 1$).

After computing each message, the pooling function $U_t$ aggregates information from the messages:

$$U_t(h_v^t, m_v^{t+1}) = h_v^t + m_v^{t+1}. \tag{5.5}$$

We selected the number of message-passing layers to be $T = 3$. Having more message-passing layers increases the receptive field and allows the model to incorporate higher-order information for each node, by including interactions between nodes that are further apart in the graph. However, having too many layers makes the model difficult to train. Empirically, we observed that using more than 3 message-passing layers made the training process unstable.

The chosen dimensions of the internal hidden representation of each node and each edge in the graph, $d_n = 64$ and $d_e = 16$, respectively, were inspired by Gilmer et al. [7]. Full hyperparameter sweeps are outside the scope of this work, but because our model was able to overfit on a small subset of the data, these hyperparameters do give our model the capacity to learn functions of this complexity.

## 5.3 Fragmentation Model Definition

We now define a model that predicts $P(\mathcal{F}_{i+1}|\mathcal{F}_i, \mathcal{M})$, the probability of obtaining fragment $\mathcal{F}_{i+1}$, given the preceding (parent) fragment $\mathcal{F}_i$ and the root molecule $\mathcal{M}$.

First, we use the message-passing neural network to build a representation at each node of $F_i$. Since the same substructure produced by different parent molecules may fragment differently, we concatenate an embedding of the root molecule $\mathcal{M}$ to the representation at each node in $\mathcal{F}_i$. To obtain this root molecule embedding, we also apply a message passing network with the same architecture to the root molecule $\mathcal{M}$, then an average pooling layer to combine the representations at each node:

$$h_x = \frac{1}{|x|} \sum_{v \in x} h_v, \tag{5.6}$$

where $|x|$ is the number of atoms in $x$ and $h_v \in \mathbb{R}^{d_n}$ is the hidden representation for each atom $v$ in $x$.

Then, we apply a linear layer to transform the representations at each node to a single value. Finally, we apply a sigmoid layer to transform the output at each node to between 0 and 1, to obtain the probability of observing a fragment resulting from the removal of that atom, $P(\mathcal{F}_{i+1}|\mathcal{F}_i, \mathcal{M})$. Mathematically,

$$\hat{y}_v = (1 + \exp(-Wc(h_v, h_{\mathcal{M}})))^{-1}, \tag{5.7}$$

where $c(h_v, h_{\mathcal{M}})$ is the concatenation of $h_v$ and $h_{\mathcal{M}}$ and $W \in \mathbb{R}^{1 \times 2d_n}$.

The loss function is binary cross-entropy loss, where

$$L(\mathcal{F}_i) = -\frac{1}{N} \sum_{v \in \mathcal{F}_i} (y_v \log(\hat{y}_v) + (1 - y_v) \log(1 - \hat{y}_v)), \tag{5.8}$$

where $N$ is the number of nodes in $\mathcal{F}_i$, $y_v$ is the true label (1 if $\mathcal{F}_i$ fragments at node $v$, 0 otherwise), and $\hat{y}_v$ is the predicted probability that $\mathcal{F}_i$ fragments at $v$.

We train the model on a single RTX A5000 NVIDIA GPU (CUDA Version 11.6), using the PyTorch Lightning library. The model is trained with a batch size of 32 and the Adam optimizer [12].

## 5.4 Fragmentation Tree Results

We evaluate model performance using two metrics: binary cross-entropy loss (Equation 5.8) and tree coverage. We define tree coverage as the fraction of fragments in the true fragmentation tree that are present in the predicted fragmentation tree:

$$\text{Tree coverage} = \frac{|\hat{\mathcal{F}} \cap \mathcal{F}|}{|\mathcal{F}|}, \tag{5.9}$$

where $\mathcal{F}$ is the set of fragments in the true fragmentation tree, and $\hat{\mathcal{F}}$ is the set of fragments in the predicted fragmentation tree.

We train a forward neural network to predict fragmentation trees from molecules, using our labeled dataset. To build the fragmentation tree, our model predicts the probability each atom in a molecule is removed. The loss curve in Figure 5.4.1 shows that the model properly learns to remove atoms to generate fragmentation trees, and generalizes beyond the training set. An example comparing a predicted tree to a true tree is shown in Figure 5.4.2.

Furthermore, our model is able to generate predicted fragmentation trees that correctly cover the fragments in the true fragmentation tree, a key desired outcome from our first tree generation module. By default, a molecule is fragmented at a specific atom if the predicted removal probability for that atom is greater than 0.5. As shown in Figure 5.4.3 and Figure 5.4.4, as the removal threshold decreases, the average coverage increases. However, as the removal threshold decreases, we also see that the number of molecule fragments in the fragmentation tree increases substantially. The default removal threshold of 0.5 appears to strike a good balance between coverage and tree size, and we will use this value going forward.

When deploying such models, it is important to understand in which settings

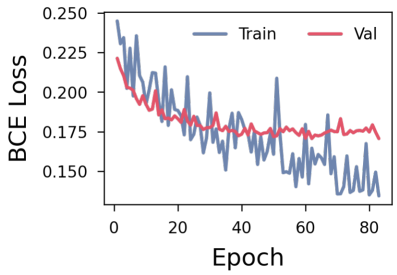


**Figure 5.4.1:** Training and validation loss curves, shown at 80 epochs with a binary cross-entropy (BCE) loss.

we can trust them. To do so, we stratify results by molecular mass and find that our model predicts fragmentation trees with better coverage for smaller molecules (molecular mass less than 600 amu) than larger molecules, as seen in Figure 5.4.5. For example, the predicted tree for the molecule $C_{14}H_{22}N_2O_3$, with a molecular mass of 266.163 amu, has a coverage of 1, which means the predicted tree includes all the subfragments in the true fragmentation tree. In contrast, for the molecule $C_{42}H_{78}N_2O_{14}$, which has a molecule mass of 834.545 amu, the predicted tree has a coverage of only 0.056.

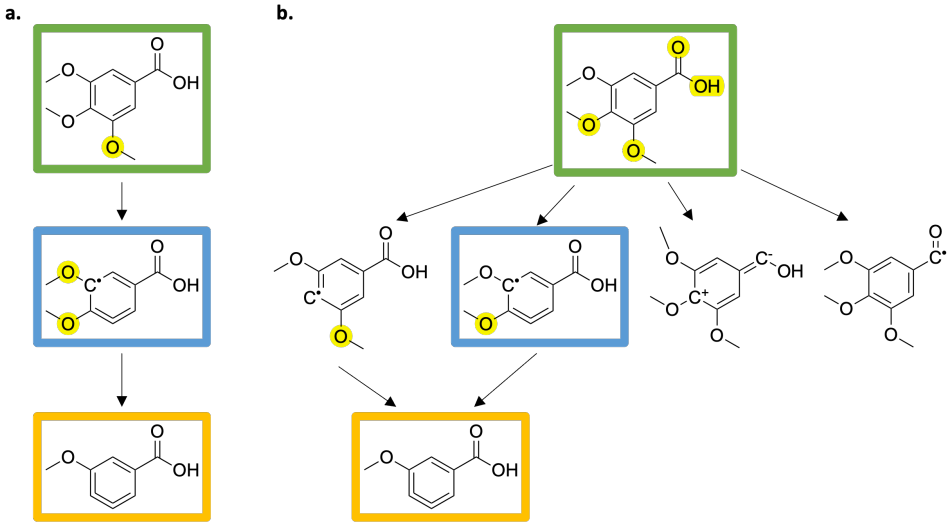


**Figure 5.4.2:** True fragmentation tree **(a)** and predicted tree **(b)** for 3,4,5-trimethoxybenzoic acid. Corresponding structures that are found in both trees are denoted with the same color. Atoms that are removed in fragmentation steps are highlighted yellow.

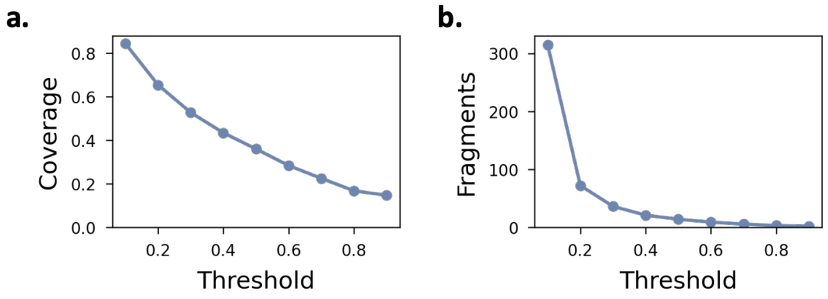


**Figure 5.4.3:** As the atom removal threshold increases, the coverage and number of fragments both decrease. **a.** Tree coverage vs. predicted likelihood threshold for atom removal. **b.** Number of fragments across various removal thresholds.

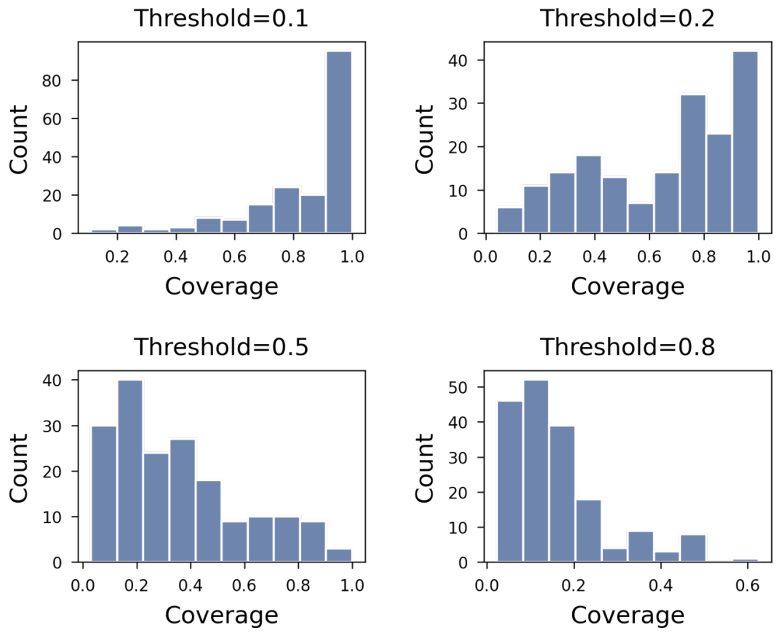


**Figure 5.4.4:** Tree coverage increases as removal threshold decreases.

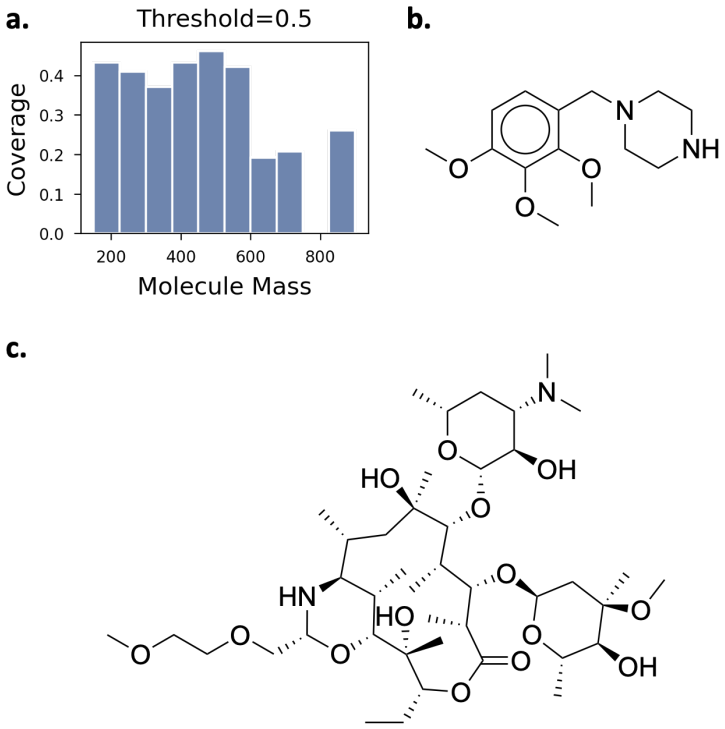


**Figure 5.4.5:** Predicted fragmentation graphs have better coverage for smaller molecules than larger molecules. **a.** Average coverage vs. binned molecular masses. **b.** Chemical structure of $C_{14}H_{22}N_2O_3$, a smaller molecule that resulted in a predicted tree with full coverage. **c.** Chemical structure of $C_{42}H_{78}N_2O_{14}$, a larger molecule that resulted in a predicted tree with a coverage of only 0.056.

# 6

# Intensity Prediction

The second task in the prediction pipeline is to predict the intensities of each fragment in the fragmentation tree from our first model.

## 6.1 Problem Formulation

We define a second neural network model to predict the intensities $i$ of each fragment $\mathcal{F}_j$. The model takes as input a pair $(\mathcal{F}_j, \mathcal{M})$, where $\mathcal{F}_j$ is a substructure in the fragmentation tree, and $\mathcal{M}$ is the root molecule in the fragmentation tree. The model outputs a tuple with 5 relative intensities $i = (i_{-2}, i_{-1}, i_0, i_1, i_2)$, where each corresponds to one of the hydrogen deviations from hydrogen rearrangements of $\mathcal{F}_j$.

## 6.2 Intensity Model Definition

First, to obtain a numerical representation of a molecule, we compute its Morgan fingerprint [17]. To do so, the circular neighborhood of each atom is hashed to produce a 2048-bit vector representation for the entire molecule. The Morgan fingerprint of a molecular graph captures the presence or absence of subgraphs centered around each node with varying radii, corresponding to the presence or absence of various subgroups centered around each atom.

We concatenate the Morgan fingerprints of $\mathcal{F}_j$ and $\mathcal{M}$, then apply a linear layer to map the representation from size 4096 to size $d$, where $d$ is the dimension of the internal hidden representation. We apply 3 feed forward blocks, where each block is constructed with a dense linear layer, non-linear activation (rectified linear unit, ReLU), and dropout. Then, we apply a 3 successive Transformer multi-head attention layers [25] to learn the relationships between multiple substructures.

Finally, we apply a linear layer to map the representation from size $d$ to size 5, as each fragment can be observed at any of 5 different hydrogen deviations, and apply a sigmoid layer to transform the output to between 0 and 1, corresponding to the relative intensity at each hydrogen deviation.

The loss function uses cosine similarity, where

$$L(\mathcal{F}_j) = 1 - \frac{\hat{y}_j \cdot y_j}{\|\hat{y}_j\|_2 \cdot \|y_j\|_2}, \tag{6.1}$$

where $\hat{y}_j$ is the predicted relative intensities and $y_j$ is the true relative intensities for fragment $\mathcal{F}_j$.

Like in the fragmentation model, we train the model on a single RTX A5000 NVIDIA GPU (CUDA Version 11.6), using the PyTorch Lightning library. We also use the Adam adaptive learning rate optimization and a minibatch of 32 samples.

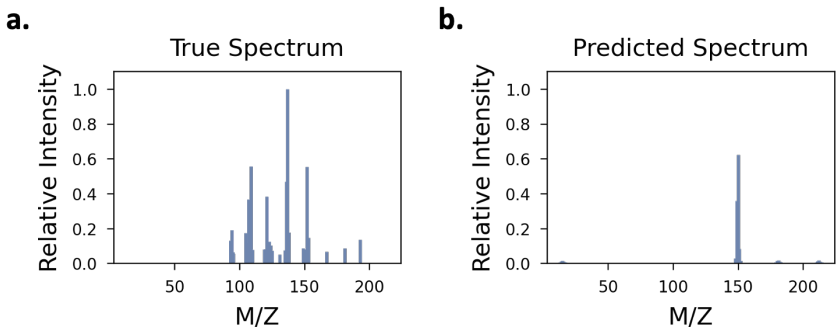


**Figure 6.3.1:** True spectrum **(a)** and predicted spectrum **(b)** for 3,4,5-trimethoxybenzoic acid.

## 6.3 Intensity Prediction Results

We train the forward neural network to predict the intensity of each fragment, using our dataset of labeled fragmentation trees. From the loss curve (Figure 6.3.2), we can see that the model fails to generalize beyond the training set. An example comparing a predicted spectrum to a true spectrum is shown in Figure 6.3.1.

We compare the performance of our model to that of NEIMS [28], a feed-forward network that takes the Morgan fingerprint of a molecule as input and outputs a mass-binned spectrum. We use two evaluation metrics: cosine similarity between the true spectra and predicted spectra, and spectrum coverage. We define spectrum coverage as the fraction of peaks in the true spectrum that are present in the predicted spectrum. We find that NEIMS outperforms our fragmentation-based model in both metrics, with higher cosine similarity and spectrum coverage (Table 6.3.1).

We believe that one reason for the poor performance is that the model is trained on the ground-truth labeled fragmentation trees, but is evaluated on the predicted fragmentation trees generated by our first model, a domain shift. The fragments in the predicted trees are a superset of the fragments in the

| | Cosine Similarity | Spectrum Coverage |
|---|---|---|
| NEIMS | 0.512 | 0.561 |
| Fragmentation-based Model | 0.289 | 0.242 |

**Table 6.3.1:** NEIMS outperforms our model in both cosine similarity loss and spectrum coverage.

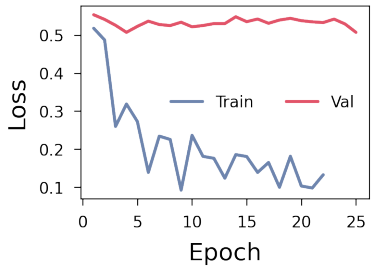


**Figure 6.3.2:** Training and validation loss curves for intensity prediction model, shown at 20 epochs with a cosine similarity loss.

ground-truth trees and will have more noise, but it is still tractable to train our intensity model on the predicted trees, instead of the ground-truth trees, and this may provide better results.

# 7 Conclusion

## 7.1 Limitations and Future Work

There are a number of avenues by which the accuracy of these methods may be improved.

First, we can train the intensity prediction model on the predicted fragmentation trees from the fragmentation model, rather than the ground-truth fragmentation trees, to address the domain shift issue.

In addition, using a less restrictive atom-removal threshold (i.e., a removal threshold lower than 0.5) in the fragmentation model may help the intensity prediction model learn more of the graph. Currently, the predicted fragmentation trees only cover around 50% of the fragments in the ground-truth fragmentation tree, on average.

## 7.2 Conclusion

With this thesis we present a framework for improving metabolite identification from mass spectrometry spectra, by learning to fragment molecular graphs. In particular, we found that our model is able to generate predicted fragmentation trees that correctly cover the fragments in the true fragmentation tree. By predicting full fragmentations, we formulate a more interpretable model compared to current molecule to spectra approaches.

While the methods described are largely introduced and motivated in the context of metabolomics, they are general to molecules analyzed with tandem mass spectroscopy, so can also be applied to other categories of molecules. Furthermore, this research expands the possibilities for deep learning applied to analytical chemistry and serves as inspiration for future domain-inspired models.

# References


[1] Felicity Allen, Russ Greiner, and David Wishart. Competitive fragmentation modeling of esi-ms/ms spectra for putative metabolite identification. *Metabolomics*, 11(1):98–110, 2015.

[2] Wout Bittremieux, Mingxun Wang, and Pieter C Dorrestein. The critical role that spectral libraries play in capturing the metabolomics community knowledge. *Metabolomics*, 18(12):94, 2022.

[3] Connor W Coley, William H Green, and Klavs F Jensen. Machine learning in computer-aided synthesis planning. *Accounts of chemical research*, 51(5):1281–1289, 2018.

[4] Katja Dettmer, Pavel A Aronov, and Bruce D Hammock. Mass spectrometry-based metabolomics. *Mass spectrometry reviews*, 26(1):51–78, 2007.

[5] Kai Dührkop, Markus Fleischauer, Marcus Ludwig, Alexander A Aksenov, Alexey V Melnik, Marvin Meusel, Pieter C Dorrestein, Juho Rousu, and Sebastian Böcker. Sirius 4: a rapid tool for turning tandem mass spectra into metabolite structure information. *Nature methods*, 16(4):299–302, 2019.

[6] Kai Dührkop, Huibin Shen, Marvin Meusel, Juho Rousu, and Sebastian Böcker. Searching molecular structure databases with tandem mass spectra

using csi: Fingerid. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 2015.

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[8] Samuel Goldman, Jeremy Wohlwend, Martin Stražar, Guy Haroush, Ramnik J. Xavier, and Connor W. Coley. Annotating metabolite mass spectra with domain-inspired chemical formula transformers. *bioRxiv*, 2022.

[9] Jürgen H Gross. *Mass spectrometry*. Springer Science & Business Media, 2006.

[10] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[11] Florian Huber, Lars Ridder, Stefan Verhoeven, Jurriaan H Spaaks, Faruk Diblen, Simon Rogers, and Justin JJ Van Der Hooft. Spec2vec: Improved mass spectral similarity scoring through learning of structural relationships. *PLoS computational biology*, 17(2):e1008724, 2021.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Frank E Koehn and Guy T Carter. The evolving role of natural products in drug discovery. *Nature reviews Drug discovery*, 4(3):206–220, 2005.

[14] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[15] Hans H Maurer. Liquid chromatography–mass spectrometry in forensic and clinical toxicology. *Journal of Chromatography B: Biomedical Sciences and Applications*, 713(1):3–25, 1998.

[16] Fred W McLafferty. Mass spectrometric analysis. molecular rearrangements. *Analytical chemistry*, 31(1):82–87, 1959.

[17] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

[18] Michael Murphy, Stefanie Jegelka, Ernest Fraenkel, Tobias Kind, David Healey, and Thomas Butler. Efficiently predicting high resolution mass spectra with graph neural networks. *arXiv preprint arXiv:2301.11419*, 2023.

[19] Louis-Félix Nothias, Daniel Petras, Robin Schmid, Kai Dührkop, Johannes Rainer, Abinesh Sarvepalli, Ivan Protsyuk, Madeleine Ernst, Hiroshi Tsugawa, Markus Fleischauer, et al. Feature-based molecular networking in the gnps analysis environment. *Nature methods*, 17(9):905–908, 2020.

[20] Mark Peplow. The robo-chemist. *Nature*, 512(7512):20, 2014.

[21] Lars Ridder, Justin JJ van der Hooft, and Stefan Verhoeven. Automatic compound annotation from mass spectrometry data using magma. *Mass Spectrometry*, 3(Special_Issue_2):S0033–S0033, 2014.

[22] Lars Ridder, Justin JJ van der Hooft, Stefan Verhoeven, Ric CH de Vos, René van Schaik, and Jacques Vervoort. Substructure-based annotation of high-resolution multistage msn spectral trees. *Rapid Communications in Mass Spectrometry*, 26(20):2461–2471, 2012.

[23] Christoph Ruttkies, Emma L Schymanski, Sebastian Wolf, Juliane Hollender, and Steffen Neumann. Metfrag relaunched: incorporating strategies beyond in silico fragmentation. *Journal of cheminformatics*, 8(1):1–16, 2016.

[24] Huibin Shen, Nicola Zamboni, Markus Heinonen, and Juho Rousu. Metabolite identification through machine learning—tackling casmi challenge using fingerid. *Metabolites*, 3(2):484–505, 2013.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[27] Mingxun Wang, Jeremy J Carver, Vanessa V Phelan, Laura M Sanchez, Neha Garg, Yao Peng, Don Duy Nguyen, Jeramie Watrous, Clifford A Kapono, Tal Luzzatto-Knaan, et al. Sharing and community curation of mass spectrometry data with global natural products social molecular networking. *Nature biotechnology*, 34(8):828–837, 2016.

[28] Jennifer N Wei, David Belanger, Ryan P Adams, and D Sculley. Rapid prediction of electron–ionization mass spectrometry using neural networks. *ACS central science*, 5(4):700–708, 2019.

[29] David S Wishart. Quantitative metabolomics using nmr. *TrAC trends in analytical chemistry*, 27(3):228–237, 2008.

[30] David S Wishart. Metabolomics for investigating physiological and pathophysiological processes. *Physiological reviews*, 99(4):1819–1875, 2019.

[31] Sebastian Wolf, Stephan Schmidt, Matthias Müller-Hannemann, and Steffen Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC bioinformatics*, 11(1):1–12, 2010.

[32] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.