Annotating Fragmentation Patterns

Sebastian Böcker[1,2], Florian Rasche[1], and Tamara Steijger[1]

[1] Lehrstuhl für Bioinformatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany
{sebastian.boecker,florian.rasche,tamara.steijger}@uni-jena.de
[2] Jena Centre for Bioinformatics, Jena, Germany

Abstract. Mass spectrometry is one of the key technologies in metabolomics for the identification and quantification of molecules in small concentrations. For identification, these molecules are fragmented by, e.g., tandem mass spectrometry, and masses and abundances of the resulting fragments are measured. Recently, methods for *de novo* interpretation of tandem mass spectra and the automated inference of fragmentation patterns have been developed. If the correct structural formula is known, then peaks in the fragmentation pattern can be annotated by substructures of the underlying compound. To determine the structure of these fragments manually is tedious and time-consuming. Hence, there is a need for automated identification of the generated fragments.

In this work, we consider the problem of annotating fragmentation patterns. Our input are fragmentation trees, representing tandem mass spectra where each peak has been assigned a molecular formula, and fragmentation dependencies are known. Given a fixed structural formula and any fragment molecular formula, we search for all structural fragments that satisfy elemental multiplicities. Ultimately, we search for a fragmentation pattern annotation with minimum total cleavage costs. We discuss several algorithmic approaches for this problem, including a randomized and a tree decomposition-based algorithm. We find that even though the problem of identifying structural fragments is NP-hard, instances based on molecular structures can be efficiently solved with a classical branch-and-bound algorithm.

1 Introduction

Mass spectrometry in combination with liquid or gas chromatography (LC-MS, GC-MS) is the most widely used high-throughput technique to analyze metabolites. Since the manual interpretation and annotation of such mass spectra is time-consuming, automated methods for this task are needed. Established methods for metabolite identification rely on comparison with a database of reference spectra. A major challenge is that most of the metabolites remain unknown: Current estimates are up to 20 000 metabolites for any given higher eukaryote [8]. Even for model organisms, only a tiny part of these metabolites has been identified. Recently, different techniques for the *de novo* interpretation of mass spectra have been proposed [12, 3, 4], but these techniques are limited to the

S.L. Salzberg and T. Warnow (Eds.): WABI 2009, LNBI 5724, pp. 13–24, 2009.

determination of molecular formulas. But only if the structure of a molecule is known, one can consider the molecule fully identified, as this structure determines chemical properties and, hence, the function of a metabolite, as well as possible interactions with proteins.

In mass spectrometry, fragmentation spectra are used to gather information about the structure of a molecule. In this work, we annotate fragment peaks in a spectrum of a *known* compound with molecular structures. Firstly, this can verify hits in a metabolite database, which are based, say, on molecular formula alone. Secondly, structurally annotated fragmentation spectra of reference compounds may help us to deduce structural information about an unknown compound. This might be achieved by comparing the fragmentation pattern of the unknown and the reference compound. An automated method for fragmentation pattern alignment has been proposed recently [5].

There exist several tools for the annotation of fragmentation spectra (Mass Frontier, MS Fragmenter), but these are usually based on predefined fragmentation rules. Rule-based systems will err if the fragmentation of a molecule differs from what has been known so far. Also, rule-based approaches usually show unsatisfactory prediction accuracy for compounds above 300 Da, as too many fragmentation pathways can explain any fragment mass. Rule-based systems fail especially at the interpretation of tandem mass spectra from LC-MS experiments, as fragmentation principles for these spectra are only partly understood.

Heinonen *et al.* [9] propose a method to annotate a tandem mass spectrum with structural fragments of a known compound: For each peak, candidate fragments are generated and ranked according to the costs of cleaving them out of the molecular structure. For single-step fragmentation, each fragment is cleaved directly from the parent molecule, whereas multistep fragmentation allows fragments to be further fragmented. Unfortunately, their Integer Linear Program for multistep fragmentation may require several days to process even a medium-sized metabolite of about 350 Da.

We propose an approach for the automated structural annotation of tandem mass spectra, combining the fragmentation model from [9] with the *de novo* interpretation of Böcker and Rasche [4]: All peaks in the tandem mass spectrum are annotated with molecular formulas, and a hypothetical fragmentation tree represents dependencies between these fragments. Here, we identify fragments of the molecular structure that fit molecular formulas in the fragmentation tree, such that fragmentation costs are minimized. This leads to the Edge-Weighted Graph Motif problem which, unfortunately, is NP-hard. We present randomized, branch-and-bound, and heuristic algorithms for its solution. The randomized algorithm is fixed-parameter tractable [11], and guarantees to find the exact solution with high probability. We limit our branch-and-bound search to a fixed maximal number of bonds that can break. Finally, we present a heuristic based on tree decomposition and dynamic programming.

Our ultimate goal is to assign substructures to all nodes of the fragmentation tree, such that *total* fragmentation costs in the tree are minimized. We propose a branch-and-bound heuristic that recursively follows the TOP-p substructures

along the fragmentation tree. We also correct errors in the underlying fragmentation tree, that stem from fragments inserted "too deep" in the tree. We have validated our methods using Orbitrap and orthogonal time-of-flight MS data. We find that each fragmentation step requires to break only a small number of bonds in the molecular structure. Despite the above hardness result, we can process molecules of masses up to 500 Da in a matter of seconds. Finally, our method allows us to validate the fragmentation tree computation from [4].

2 Preliminaries

Fragmentation spectra of molecules can be measured with different experimental setups. Here, we concentrate on collision-induced dissociation (CID), where ions collide with a neutral gas. The overall size of fragments can be adjusted using the collision energy. If the sample is separated by mass twice, once before and once after fragmentation, one speaks of tandem mass spectrometry. For peptide and glycan fragmentation, the experimental setup is chosen such that the fragmentation happens mainly at the peptide and glycosidic bonds. In contrast, fragmentation of metabolites is rather unpredictable. In the following, we will not use prior information about metabolite fragmentation such as fragmentation rules, with the exception of edge weights that represent fragmentation energies.

Böcker and Rasche [4] propose an approach for the determination of molecular formulas from tandem mass spectra. They construct a fragmentation graph where vertices represent all molecular formulas that can explain a peak in the fragmentation spectrum, and edges represent possible fragment relationships. Finding a fragmentation tree in this graph leads to the NP-hard Maximum Colorful Subtree problem. Using the exact algorithms from [4], fragmentation trees often resemble trees constructed manually by an expert, and can be computed in seconds. Vertices in the fragmentation tree are labeled by the hypothetical molecular formula of the corresponding fragment. See Fig. 1 for an example. We will use such vertex-labeled fragmentation trees as input for our approach. The method may insert fragments too low in the fragmentation tree, because the molecular formulas of fragments allow to do so, increasing the score of the fragmentation tree. There is no way to tackle this methodical problem unless the molecular structure of the compound is known.

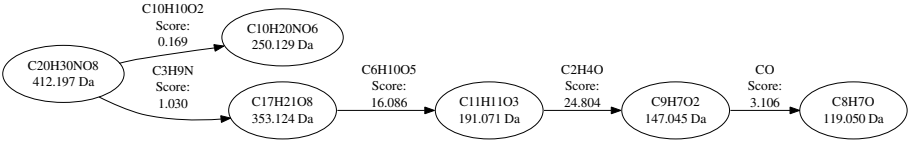

Fig. 1. The fragmentation tree of hexosyloxycinnamoyl choline as calculated by exact algorithms from [4]. Nodes are labeled with the molecular formula of the fragment and the mass of the peak. Edge labels consist of the neutral loss and a score that represents the likelihood that the corresponding fragmentation reaction is real.

3 Formal Problem Definition

We use two different models for the fragmentation process: One-step fragmentation and multistep fragmentation [9]. Note, that single-step fragmentation [9] is a special case of one-step fragmentation, where all fragments are cleaved from the parent molecule.

Let Σ be the alphabet of elements in our molecules, such as $\Sigma = \{\mathrm{C}, \mathrm{H}, \mathrm{N}, \mathrm{O}, \mathrm{P}, \mathrm{S}\}$. A *molecular structure* M consists of a simple, undirected, connected graph where all vertices are labeled with elements from Σ, and edges are weighted by positive weights $w(e) > 0$. The elements of Σ will be called *colors* in this context. The *molecular formula* indicates how many vertices of each color are present in a molecular structure, e.g., $\mathrm{C_{20}H_{30}NO_8}$. For *one-step fragmentation*, we are given a molecular structure M and a molecular formula f over Σ, and we try to find a connected subgraph of M that can be cleaved out with minimum costs, that is, minimum sum of energies for all cleaved bonds, and that has colors corresponding to f.

Edge-Weighted Graph Motif problem. Given a vertex-colored edge-weighted graph $G = (V, E)$ and a multiset of colors C of size k, find a connected subgraph $H = (U, F)$ of G such that the multiset of colors of U equals C, and H has minimum weight $w(H) := \sum_{\{u,v\} \in E, u \in U, v \in V \setminus U} w(\{u, v\})$.

This problem is a generalization of the Graph Motif problem, where no edge weights exist, and one asks whether any such subgraph exists. This problem is NP hard even for bipartite graphs of bounded degree and two colors [7]. Betzler *et al.* [2] developed a randomized FPT algorithm for the Graph Motif problem using color-coding [1]: Solving the problem with error probability ϵ takes $O(|\log \epsilon| \cdot 4.32^k \cdot k^2 \cdot |E|)$ time.

However, fragmentation pathways can consist of consecutive fragmentation steps [9], where fragments can be cleaved from other fragments. Fragmentation pathways can be represented by *fragmentation trees*, directed trees where the root corresponds to the parent molecule and each edge represents a fragmentation step. Each node n is labeled with a molecular formula that has to be a sub-formula of the molecular formula attached to n's parent.

For the *multistep fragmentation* model, we are given a molecular structure M and a fragmentation tree T. We want to assign sub-structures to the nodes of the fragmentation tree that match their molecular formulas, such that the total cost of cutting out the substructures, over all edges of the fragmentation tree, is minimized. Clearly, it does not suffice to search for the optimal graph motif in every fragmentation step independently, since following fragmentation steps may be cleaved from a suboptimal substructure with lower total costs.

In order to find a fragmentation process consistent with the given fragmentation tree, we use a search tree. Since it does not suffice to take the fragment with minimum costs in every fragmentation step, our heuristic allows the user to specify a number p so that in every step, the best p fragments are considered. For each such fragment, we build up the search tree recursively, and accept the fragment that results in lowest total costs. Finally, we check whether moving

up a node in a fragmentation tree by one level, will decrease the total cost of fragmentation. To do so, we compare the total costs of cleaving fragment f and all subsequent fragments from its parent, with the total costs of cleaving them from its grandfather.

We have noted that using methods from [4], some fragments may be attached too deep in the fragmentation tree. Following this line of thought, we can decide that the fragmentation tree cannot be trusted and should be re-computed in our optimization, comparable to the problem considered in [9]. In order to find the multistep fragmentation with the lowest total costs, we have to build up a fragmentation graph representing all possible fragmentation pathways, we omit the details. For small molecules, this can be solved using exhaustive enumeration [9]. But finding the fragmentation process with lowest costs leads to the Minimum Colorful Subtree problem which, again, is known to be NP-hard [4]. In addition, building the fragmentation graph requires to solve a huge number of Edge-Weighted Graph Motif instances. In view of this seemingly inevitable complexity, we refrain from attacking this more general problem.

4 Algorithms

In the following, we describe three algorithms to solve the Edge-Weighted Graph Motif problem. For the multistep fragmentation search tree, our algorithms also have to calculate suboptimal solutions.

4.1 Random Separation

Cai *et al.* [6] proposed a randomized technique called *random separation* based on color-coding [1]. The key idea of random separation is to partition the vertices by coloring them randomly with two different colors. Then, connected components are identified and appropriate components are tested for optimality. Random separation can be used to solve a wide range of fixed-parameter tractable problems, especially when the input graph has bounded degree. This is the case for molecular structures, where vertex degrees are bounded by the valences of elements.

We now apply random separation to the Edge-Weighted Graph Motif problem. Let k be the cardinality of the color multiset C. We search for a substructure $H = (U, F)$ that minimizes $w(H)$, where $|U| = k$. Let $N(U)$ denote the neighborhood of U in G. Given a graph $G = (V, E)$ and a random separation of G that partitions V into V_1 and V_2, there is a $2^{-(k+|N(U)|)+1}$ chance that U is entirely in V_1 and its neighborhood $N(U)$ is entirely in V_2 or vice versa. We use depth-first search to identify the connected components in V_1 and V_2. Simultaneously, colors are counted and costs for the partition are calculated. If the colors of a connected component correspond to the colors of the given multiset C and the costs are smaller than the costs of the best solution so far, the connected component is stored. In order to find the optimal solution with error probability ϵ, the procedure has to be repeated $\lceil |\log \epsilon| \,/\, |\log(1 - 2^{-(k+kd)+1})| \rceil$ times, where d is the maximum vertex degree in G.

We now analyze the worst-case running time of this approach: Coloring takes $O(|V|)$ time. Depth-first search has a running time of $O(|V| + |E|)$ but since molecular structures have bounded degree, the overall running time of one trial is $O(|V|)$. Accordingly, the overall running time of the random separation algorithm is $O(|\log \epsilon|\, 2^{(k+kd)} \cdot |V|)$. Recall that d is bounded in molecular structures. Also note that the term kd is due to the neighborhood of U in G. In our experiments, we observe that one-step fragmentation usually requires only few bonds to break. In this case, we can substitute the worst case estimation kd with maximal number b of bonds breaking in a fragmentation step. In our implementation, b is an input parameter that can be used to reduce the number of trials and, hence, to decrease running time. Obviously, b has to be chosen large enough to guarantee that, with high probability, the optimal solution is found.

4.2 Branch-and-Bound

The second algorithm is a classical branch-and-bound algorithm. It branches over edge sets that might break during a fragmentation step. Given an edge set, its deletion might separate G into a set of connected components. Similar to the random separation approach, depth-first search is used to identify components that might be selected as a solution. If a solution has been found, its costs are used as an upper bound for pruning. The user can specify the maximum number of bonds b that may break during one single fragmentation step. We then try to cut out a solution with exactly $b' = 1, \ldots, b$ edges.

Since the costs of a solution correspond to the sum of weights of deleted edges, it is not necessary to iterate over all possible edge sets. To efficiently traverse the search tree, we use an edge set iterator that avoids edge sets with too high costs. Edges are sorted in increasing order with respect to their weight. Now, we can easily iterate over all edge sets of a fixed cardinality such that the edge sets have increasing costs. Thus, as soon as a solution with b' edges has been found, or the costs exceed that of the best solution found so far, all following edge sets of the same cardinality will have higher costs and can be omitted.

Sorting edges costs $O(|E| \log |E|)$ time. Running time of the depth-first search is $O(|V|)$, as explained for random separation. The branch-and-bound algorithm iterates over $O(|V|^b)$ edge sets. This results in an overall running time of $O(|E| \log |E| + |V|^b)$. Unfortunately, running time is exponentially in b. But if the number of bonds that break in one single fragmentation step is small and bounded, b can be assumed as a constant and hence, the algorithm can be executed in polynomial time.

4.3 Tree Decomposition-Based Algorithm

Fellows *et al.* [7] show that the Graph Motif problem is already NP-hard on trees with maximum vertex degree three, assuming an unbounded alphabet. If, however, the alphabet is bounded, then the Graph Motif problem can be solved in polynomial time for graphs with bounded treewidth [7]. Unfortunately, running times of this algorithm are too high for applications. Similarly, there is a

polynomial time algorithm to solve the Edge-Weighted Graph Motif problem on trees, we defer the details to the full paper. We now adapt ideas from this algorithm for a heuristic algorithm, that uses the concept of tree decomposition to solve the Edge-Weighted Graph Motif problem on arbitrary graphs.

A *tree decomposition* of a graph $G = (V, E)$ is a pair $\langle \{X_i \mid i \in I\}, T \rangle$ where each X_i is a subset of V, called a *bag*, and T is a tree containing the elements of I as nodes, satisfying: a) $\bigcup_{i \in I} X_i = V$; b) for every edge $\{u, v\} \in E$, there is an $i \in I$ such that $\{u, v\} \subseteq X_i$; and c) for all $i, j, h \in I$, if j lies on the path between i and h in T then $X_i \cap X_h \subseteq X_j$. Let $\omega := \max\{|X_i| \mid i \in I\}$ be the maximal size of a bag in the tree decomposition. The *width* of the tree decomposition equals $\omega - 1$. The *treewidth* of G is the minimum number $\omega - 1$ such that G has a tree decomposition of width $\omega - 1$.

To simplify the description and analysis of our algorithm, we use nice tree decompositions in the remainder of this paper. Here, we assume the tree T to be arbitrarily rooted. A tree decomposition is a *nice* tree decomposition if every node of the tree has at most two children, and there exist only three types of nodes in the tree: A *join node* i has two children j and h such that $X_i = X_j = X_h$. An *introduce node* i has only one child j, and $|X_i| = |X_j| + 1$ as well as $X_j \subset X_i$ holds. A *forget node* i has only one child j, and $|X_i| = |X_j| - 1$ as well as $X_i \subset X_j$ holds. Using methods from [10], we can transform a tree decomposition with m nodes into a nice tree decomposition with the same width and $O(m)$ nodes in linear time. Finally, for each leaf X_i with $|X_i| > 1$ we can insert additional introduce nodes under X_i, such that the new leaf contains only a single vertex.

Assume that a nice tree decomposition $\langle \{X_i \mid i \in I\}, T \rangle$ of width $\omega - 1$ and $O(m)$ nodes of the molecule graph $M = (V, E)$ is given. In the following, we describe a dynamic programming heuristic to solve the Edge-Weighted Graph Motif problem using the nice tree decomposition of the molecule graph. Let Y_i be the vertices in V that are contained in the bags of the subtree below node i. Let $c(v)$ be the color of a vertex $v \in V$ in the molecule graph, and let $c(U)$ be the *multiset* of colors for a vertex set $U \subseteq V$. We define costs for $U \subseteq V$ and color multiset C as $costs(U, C) := w(U)$ if $c(U) = C$, and $costs(U, C) := \infty$ otherwise. For each node i of the tree decomposition, we want to calculate the costs $W_i(C, U)$ for building up a fragment using the multiset of colors C and vertices $U \subseteq X_i$. These values are stored in a table and calculated using dynamic programming. Our algorithm starts at the leaves of the tree decomposition. For each leaf i with $X_i = \{v\}$ we initialize

$$W_i\big(\{c(v)\}, \{v\}\big) = w\big(\{v\}\big).$$

During the bottom-up traversal, the algorithm distinguishes if i is an introduce node, a forget node, or a join node. We now give recurrences to compute the matrix $W_i(C, U)$ where $U \subseteq X_i$ and $c(U) \subseteq C$. For readability, we omit the condition that all solutions need to be connected from the recurrences. This has to be checked in addition for all introduce and join nodes.

Forget nodes. These nodes result in a simple recurrence, so we treat them first: Let i be the parent node of j and choose v with $X_j \setminus X_i = \{v\}$. Then,

$$W_i(C,U) = \min\bigl\{W_j(C,U), W_j(C,U \cup \{v\}), \mathit{costs}(U,C)\bigr\}.$$

Join nodes. Let i be the parent node of j and h, where $X_i = X_j = X_h$. We want to compute $W_i(C,U)$. To do so, we iterate over all $U_1, U_2 \subseteq U$ such that $U_1 \cup U_2 = U$. Let $C' = C \setminus c(U)$ be the multiset of remaining colors. We then iterate over all bipartitions C'_1, C'_2 of C', that is, $C'_1 \cup C'_2 = C'$ and $C'_1 \cap C'_2 = \emptyset$. Let $C_1 := c(U_1) \cup C'_1$ and $C_2 := c(U_2) \cup C'_2$. We now access the values $W_j(C_1, U_1)$ and $W_h(C_2, U_2)$ that represent minimal cost for the respective instances. Using our traceback matrix, we backtrace through the tree decomposition and reconstruct the sets of vertices $V_1 \subseteq Y_j$, $V_2 \subseteq Y_h$ used in the corresponding optimal solutions with weights $W_j(C_1, U_1)$ and $W_h(C_2, U_2)$. If $V_1 \cap V_2 \not\subseteq U$ then we stop, as the partial solutions overlap outside of the current bag X_i. Otherwise, we can compute $\mathit{costs}(V_1 \cup V_2, C)$. We take the minimum of all these values as $W_i(C,U)$.

Introduce nodes. Let i be the parent node of j and choose v with $X_i \setminus X_j = \{v\}$. We distinguish two different situations: If v is the last vertex of a cycle in M, so that v *closes a cycle*, then we have to compute $W_i(C,U)$ using a traceback through the tree decomposition, combining two optimal solutions. This is analogous to the case of a join node, we defer the details to full paper. Otherwise, we set

$$W_i(C,U) = \begin{cases} \min\bigl\{W_j(C \setminus \{c(v)\}, U \setminus \{v\}), \mathit{costs}(U,C)\bigr\} & \text{if } v \in U, \\ \min\{W_j(C,U), \mathit{costs}(U,C)\} & \text{otherwise.} \end{cases}$$

The minimum costs to cleave a fragment from M are $\min_{i, U \subseteq X_i} W_i(C,U)$. The corresponding fragment can be computed by a traceback.

We can close a cycle either by introducing the last vertex of it, or by joining two parts of a cycle. In this case, we cannot guarantee that the above recurrences result in an optimal solution: Optimal partial solutions in a cycle might overlap and be discarded, whereas suboptimal, non-overlapping partial solutions may be used to build up an optimal solution but are not stored, as we limit calculations to optimal solutions at all times. In order to check for connectivity the algorithm needs to perform tracebacks through the matrices W_i. To achieve feasible running times, use of an explicit traceback matrix was inevitable. The drawback is that the tree decomposition heuristic has only limited support for finding suboptimal solutions, severely limiting its use for multistep fragmentation.

The algorithm needs $O(m \cdot |\Sigma|^k \cdot 2^\omega)$ memory in the worst case. Forget nodes and introduce nodes that do not close a cycle can be calculated in $O(\omega \cdot d)$ while join nodes and the remaining introduce nodes need $O(m \cdot k \cdot 2^k \cdot 3^\omega)$. Running time of the algorithm is $O(m \cdot |\Sigma|^k \cdot 2^\omega \cdot (m \cdot k \cdot 2^k \cdot 3^\omega + \omega \cdot d))$, we defer the slightly technical details of the running time analysis to the full paper.

The molecular formulas in X_i are restricted by the available elements of f and further by the elements of the vertices in Y_i, so the number of molecular

formulas that have to be checked for each node i is significantly smaller than what worst-case running times suggest. We use a hash map to store and access only those entries of the matrices $W_i(C, U)$ that are smaller than infinity. This reduces both running times and memory of our approach in applications.

5 Experimental Results

We implemented our algorithms in Java 1.5. Running times were measured on an Intel Core 2 Duo processor, 2.5 GHz with 3 GB memory. To compute the tree decomposition of the molecule graphs, we used the method QuickBB in the library LibTW implemented by van Dijk *et al.* (`http://www.treewidth.com`). We implemented a method to transform the computed tree decomposition into a nice tree decomposition. For the random separation algorithm, we use an error probability $\epsilon = 0.1\%$, so that the optimal solution will be found with a probability of 99.9%. In the multistep fragmentation evaluation, we set $p = 5$, thus, keeping the five best substructures in each fragmentation step.

As test data we used 35 fragmentation trees calculated from CID mass spectra measured on an API QStar Pulsar Hybrid instrument (Applied Biosystems) and 8 trees from an LTQ Orbitrap XL instrument (Thermo Fisher Scientific) using PQD fragmentation. The test compounds consisted of biogenic amino acids, complex choline derivatives, and commercially available pharmacological agents. QStar data is publicly available from `http://msbi.ipb-halle.de/MassBank`.

Since hydrogen atoms are often subject to rearrangements, we do not include them in our calculations. We do, however, support some minor structural rearrangements such as hydroxyl group rearrangements. These occur frequently as a result of cyclizations. We model this using pseudo-edges. Our model is not biochemically correct but enables us to reconstruct fragmentation trees with minor rearrangements, e.g., the fragmentation tree of arginine where a hydroxyl group is rearranged because of cyclization.

Detailed information about running times of the multistep heuristic using the different approaches can be found in Table 1. One can see that the branch-and-bound

Table 1. The average running times of the algorithms: neighborhood for random separation has been estimated with $b = 5$, branch-and-bound allowed $b = 3$ (BB-3) and $b = 5$ (BB-5) bonds to break, running time of tree decomposition-based algorithm includes computing the tree decomposition itself. Multistep fragmentation considered the 5 best fragments in every step.

		multistep heuristic			
Mass (Da)	#comp.	RS	BB-3	BB-5	TD
< 100	1	< 1 s	< 1 s	< 1 s	< 1 s
100–200	17	23.2 s	0.1 s	0.2 s	1.1 s
200–300	16	50.7 min	0.7 s	6.0 s	13.3 s
300–400	3	4.8 h	6.7 s	55.7 s	49.2 s
400–500	5	> 1 day	0.5 s	5.6 s	1.7 min
> 500	1	> 1 week	10.2 min	4.6 h	2.0 h

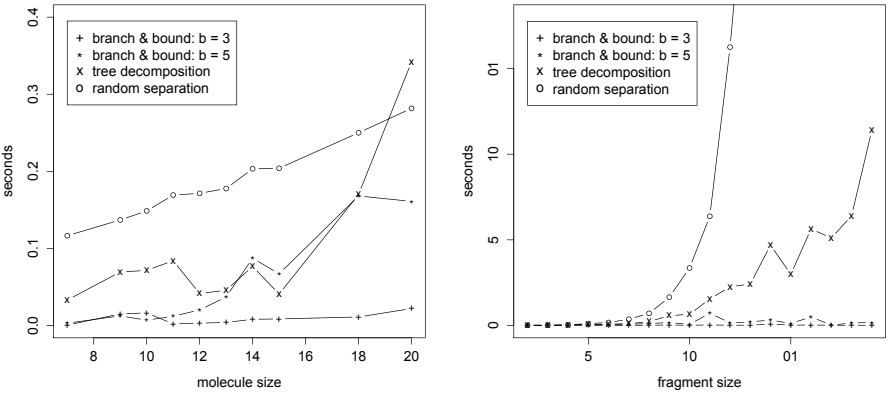

Fig. 2. Running time comparison of the three algorithms: The left diagram shows the average running time depending on the molecule size $|M|$ given a fixed fragment size of six. In the right diagram the average running time for several molecule sizes in dependence on the fragment's size is displayed.

algorithms outperforms the two more sophisticated algorithms. The running time of the tree decomposition algorithm grows exponentially, but all test instances could be calculated in a feasible amount of time. The random separation algorithm, however, performs fast for small instances, but requires several days for molecules > 400 Da.

Fig. 2 shows how the running times of the algorithms depend on the size of the molecular structure M and on the size of the fragments. It illustrates that the running time of the branch-and-bound algorithm mainly depends on the size of M, particularly for larger b, whereas that of the tree decomposition algorithm depends both on fragment size and size of the molecular structure. Finally, running time of the random separation algorithm depends mainly on fragment size.

For all instances that finished computation using the random separation algorithm, we reach identical total costs as for the branch-and-bound algorithm. Annotations differ only marginally in the sequence of cleavages. The annotations found by the branch-and-bound algorithm with $b = 3$ and $b = 5$ also have identical costs. This supports our assumption that instances based on molecular graphs do not resemble the full complexity of the Edge-Weighted Graph Motif problem.

The heuristic algorithm failed to annotate 3 fragmentation trees consistently, and returned a clearly suboptimal solution in 11 cases due to its limited support for suboptimal partial solutions. Often, a suboptimal fragment that split up a ring had to be chosen early in the fragmentation tree in order to reduce subsequent costs.

Our annotations turned out to be valuable to validate the fragmentation trees proposed by methods in [4]. Our analysis of the annotated fragmentation trees identified peaks in several fragmentation trees that were annotated with a molecular formula but probably are noise peaks. These peaks have a low intensity and are also scored very low. In our analysis, we were unable to assign a fragment to the molecular formula. For example, the 250 Da node in Figure 1 was identified as noise peak. The score of the corresponding fragmentation step is very low compared to the others, and a fragment with formula $C_{10}H_{20}NO_6$ cannot be cleaved from the structure of hexosyloxycinnamoyl choline without major rearrangements. We also identified an intense peak that could not be annotated with any fragment. Consultation with an expert resulted in the conclusion that the spectrum was contaminated.

Furthermore, we identified three nodes in two fragmentation trees that had been inserted too low into the fragmentation tree, and pulling them up one level resulted in a fragmentation pattern with significantly decreased total costs. In two other fragmentation trees, we identified nodes where pulling-up results in slightly reduced costs. A closer look at the fragmentation patterns revealed that in these cases, two competitive paths might co-occur.

6 Conclusion

We have presented a branch-and-bound heuristic for the multistep fragmentation problem that aims at annotating fragmentation cascades. As a sub-task, this heuristic repeatedly needs to solve the NP-hard Edge-Weighted Graph Motif problem. We proposed a randomized FPT-algorithm, an exact branch-and-bound algorithm, as well as a heuristic to solve the problem. Our experimental results reveal that despite its theoretical complexity, real world instances of the problem can be solved quickly, as only few bonds break in each fragmentation step.

We find that the branch-and-bound search outperforms the two more involved algorithms. In case a large number of bonds has to be broken, this might be an indication of either structural rearrangements, or errors in the fragmentation tree. Thus, in those cases an experimentalist should manually annotate the fragmentation pattern. Our method was able to correct a few errors in the fragmentation trees computed in [4], making alignments of those trees, as proposed in [5], more reliable.

In the future, we want to improve the support for structural rearrangements, e.g., by letting the user introduce pseudo-edges in the molecule which represent expected rearrangements. Furthermore, the cost function can be improved by taking into consideration aromatic bonds, and by updating the valences after each fragmentation step. Additionally, we plan to integrate fragmentation rules into our model through the modification of edge weights. To improve the annotation reliability when using the multistep fragmentation model, intermediate concepts, such as two-step fragmentation, might be helpful. Since our method is "model-free" it can, in principle, also be applied to other fragmentation techniques.

Acknowledgments. We thank the department of Stress and Developmental Biology at the Leibniz Institute of Plant Biochemistry in Halle, Germany and Aleš Svatoš from the Max Planck Institute for Chemical Ecology in Jena, Germany for supplying us with the metabolite mass spectra, and Christoph Böttcher for the manual identification of fragmentation trees.

References

1. Alon, N., Yuster, R., Zwick, U.: Color-coding. J. ACM 42(2), 844–856 (1995)
2. Betzler, N., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithms and hardness results for some graph motif problems. In: Ferragina, P., Landau, G.M. (eds.) CPM 2008. LNCS, vol. 5029, pp. 31–43. Springer, Heidelberg (2008)
3. Böcker, S., Letzel, M., Lipták, Z., Pervukhin, A.: SIRIUS: Decomposing isotope patterns for metabolite identification. Bioinformatics 25(2), 218–224 (2009)
4. Böcker, S., Rasche, F.: Towards de novo identification of metabolites by analyzing tandem mass spectra. Bioinformatics 24, I49–I55 (2008); Proc. of European Conference on Computational Biology (ECCB 2008)
5. Böcker, S., Zichner, T., Rasche, F.: Automated classification of unknown biocompounds using tandem MS. In: Poster Proc. of Conference of the American Society for Mass Spectrometry (ASMS 2009), p. W690 (2009)
6. Cai, L., Chan, S.M., Chan, S.O.: Random separation: a new method for solving fixed-cardinality optimization problems. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 239–250. Springer, Heidelberg (2006)
7. Fellows, M., Fertin, G., Hermelin, D., Vialette, S.: Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 340–351. Springer, Heidelberg (2007)
8. Fernie, A.R., Trethewey, R.N., Krotzky, A.J., Willmitzer, L.: Metabolite profiling: from diagnostics to systems biology. Nat. Rev. Mol. Cell Biol. 5(9), 763–769 (2004)
9. Heinonen, M., Rantanen, A., Mielikäinen, T., Kokkonen, J., Kiuru, J., Ketola, R.A., Rousu, J.: FiD: a software for ab initio structural identification of product ions from tandem mass spectrometric data. Rapid Commun. Mass Spectrom. 22(19), 3043–3052 (2008)
10. Kloks, T.: Treewidth, Computation and Approximation. Springer, Heidelberg (1994)
11. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2006)
12. Rogers, S., Scheltema, R.A., Girolami, M., Breitling, R.: Probabilistic assignment of formulas to mass peaks in metabolomics experiments. Bioinformatics 25(4), 512–518 (2009)