Computing Fragmentation Trees from Metabolite Multiple Mass Spectrometry Data

Kerstin Scheubert[1], Franziska Hufsky[1,2], Florian Rasche[1], and Sebastian Böcker[1]

[1] Lehrstuhl für Bioinformatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, Jena, Germany, {kerstin.scheubert,franziska.hufsky, florian.rasche,sebastian.boecker}@uni-jena.de
[2] Max Planck Institute for Chemical Ecology, Beutenberg Campus, Jena, Germany

Abstract. Since metabolites cannot be predicted from the genome sequence, high-throughput *de-novo* identification of small molecules is highly sought. Mass spectrometry (MS) in combination with a fragmentation technique is commonly used for this task. Unfortunately, automated analysis of such data is in its infancy. Recently, fragmentation trees have been proposed as an analysis tool for such data. Additional fragmentation steps (MS^n) reveal more information about the molecule.
We propose to use MS^n data for the computation of fragmentation trees, and present the Colorful Subtree Closure problem to formalize this task: There, we search for a colorful subtree inside a vertex-colored graph, such that the weight of the transitive closure of the subtree is maximal. We give several negative results regarding the tractability and approximability of this and related problems. We then present an exact dynamic programming algorithm, which is parameterized by the number of colors in the graph and is swift in practice. Evaluation of our method on a dataset of 45 reference compounds showed that the quality of constructed fragmentation trees is improved by using MS^n instead of MS^2 measurements.

This is a preprint of:
Kerstin Scheubert, Franziska Hufsky, Florian Rasche and Sebastian Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. In Proc. of Research in Computational Molecular Biology (RECOMB 2011), volume 6577 of Lect Notes Comput Sci, pages 377-391. Springer, Berlin, 2011.

1 Introduction

The phenotype of an organism is strongly determined by the small chemical compounds contained in its cells. These compounds are called metabolites; their mass is typically below 1000 Da. Unlike biopolymers such as proteins and glycans, the chemical structure of metabolites is not restricted. This results in a great variety and complexity in spite of their small size. Except for primary metabolites directly involved in growth, development, and reproduction, most metabolites

remain unknown. Plants, filamentous fungi, and marine bacteria synthesize huge numbers of secondary metabolites, and the number of metabolites in any higher eukaryote is currently estimated between 4 000 and 20 000 [9]. Unlike for proteins, the structure of metabolites usually cannot be deduced by using genomic information, except for very few metabolite classes like polyketides.

Mass spectrometry (MS) is one of the key technologies for the identification of small molecules. Identification is usually achieved by fragmenting the molecule, and measuring masses of the resulting fragments. The fragmentation mechanisms of electron ionization (EI) during gas chromatography MS (GC-MS) are well described [12]. Unfortunately, only thermally stable and volatile compounds can be analyzed by this technique. Liquid chromatography MS (LC-MS) can be adapted to a wider array of (even thermally unstable) molecules, including a range of secondary metabolites [9]. LC-MS uses the more gentle electrospray ionization (ESI) and a selected compound is fragmented in a second step using collision-induced dissociation (CID), resulting in MS^2 spectra. Different from peptides where CID fragmentation is generally well understood, this understanding is in its infancy for metabolites. The manual interpretation of CID mass spectra is cumbersome and requires expert-knowledge. Even searching spectral libraries is problematic, since CID mass spectra are limited in their reproducibility on different instruments [14]. Additionally, compound libraries to search against are vastly incomplete. For these reasons, automated *de novo* interpretation of CID mass spectra is required as an important step towards the identification of unknowns.

Multiple-stage mass spectrometry (MS^n) allows to further fragment the products of the initial fragmentation step. To this end, fragments of the MS^2 fragmentation are selected as precursor ions, and subjected to another fragmentation reaction. Several precursor ions can be selected successively. Selection can either be performed automatically for a fixed number of precursor ions with maximal intensity, or manually by selecting precursor ions. Fragments from MS^3 fragmentations can, in turn, again be selected as precursor ions, resulting in MS^4 spectra. Typically, the quality of mass spectra is reduced with each additional fragmentation reaction. Furthermore, measuring time is increased, reducing the throughput of the instrument. Hence, for untargeted analysis by LC-MS, analysis is usually limited to few additional fragmentation reactions beyond MS^2.

In the past years some progress has been made in searching of spectral and compound libraries using CID spectra [11,14,15], and there exist some pioneering studies towards the automated analysis of such spectra [10, 16, 18]. Recently, a method for *de novo* interpretation of metabolite MS^2 data has been developed [6, 17]. It helps to identify metabolite sum formulas and further to interpret the fragmentation processes, resulting in hypothetical fragmentation trees. These fragmentation trees can be compared to each other to identify compound classes of unknowns [17]. In fact, applying this method of computing fragmentation trees to MS^n data is possible, but dependencies between different fragmentation steps are not taken into account. For peptide sequencing, MS^3 spectra have been used to increase the accuracy of *de novo* peptide sequencing algorithms [2].

Here, we present a method for automated interpretation of MS^n data. We adjust the fragmentation model for MS^2 data from [6] to MS^n data to reflect the succession of fragmentation reactions. This results in the Colorful Subtree Closure problem that has to be solved in conjunction with the original Maximum Colorful Subtree problem [6]. We show that the Colorful Subtree Closure problem is NP-hard, and present intractability results regarding the approximability of this and the Maximum Colorful Subtree problem. Despite these negative results, we present an exact algorithm for the combined problem: This fixed-parameter algorithm, based on dynamic programming, has a worst-case running time with exponential dependence only on the number of peaks k in the spectrum. In application, we choose some fixed k' such as $k' = 15$, limit exact calculations to the k' most intense peaks in the mass spectra and attach the remaining peaks heuristically. We apply our algorithm to a set of 185 mass spectra from 45 compounds, and show that adding MS^n information to the analysis improves quality of results but does not affect the running time in comparison to the algorithm for MS^2 data from [6].

2 Constructing Fragmentation Trees from MS^2 and MS^n Data

Fragmentation of glycans and proteins is generally well understood, but this is not the case for metabolites and small molecules. That makes it difficult both to predict the fragmentation process, and to interpret metabolite MS data. Böcker *et al.* [6] propose fragmentation trees to interpret MS^2 data: In a fragmentation tree nodes are annotated with molecular formulas of fragments, and edges represent fragmentation reactions or *neutral losses*.

The algorithm to compute a fragmentation tree proceeds as follows [6]: Each fragment peak is assigned one or more molecular formulas with mass sufficiently close to the peak mass [5]. The resulting molecular formulas including the parent molecular formula, are considered vertices of a directed acyclic graph (DAG). We assume that the parent molecular formula is either given or can be calculated from isotope pattern analysis. Vertices in the graph are colored, such that vertices that explain the same peak receive the same color. Edges represent neutral losses, that is, fragments of the molecule that are not observed, as they were not ionized. Two vertices u, v are linked by a directed edge if the molecular formula of v is a sub-molecule of the molecular formula of u. Edges are weighted, reflecting that some edges are more likely to represent true neutral losses than others. Also, peak intensities and mass deviations are taken into account in these weights. Now, each subtree of the resulting graph corresponds to a possible fragmentation tree. To avoid the case that one peak is explained by more than one molecular formula, only *colorful* subtrees that use every color at most once are considered. In practice, it is very rare that a peak is indeed created by two different fragments, whereas our optimization principle without restriction would always choose all explanations of a peak. Therefore, searching for a

colorful subtree of maximum weight means searching for the best explanation of the observed fragments:

Maximum Colorful Subtree problem. Given a vertex-colored DAG $G = (V, E)$ with colors $\mathcal{C}$ and weights $w : E \to \mathbb{R}$. Find the induced colorful subtree $T = (V_T, E_T)$ of G of maximum weight $w(T) := \sum_{e \in E_T} w(e)$.

We now modify this problem to take into account MS^n data when constructing fragmentation trees. From the experimental data, we construct a DAG $G = (V, E)$ together with a vertex coloring $c : V \to \mathcal{C}$, called *fragmentation graph*. Recall that the vertices of V correspond to potential molecular formulas of the fragments, colors $\mathcal{C}$ correspond to peaks in the mass spectra, and molecular formulas corresponding to the same peak mass have the same color. In contrast to the fragmentation graph where each edge indicates a direct succession, the MS^n data does not only hint to direct but also to indirect successions. So, in the graph constructed from the MS^n data we also have to score the transitive closure of the induced subtrees. The *transitive closure* $G^+ = (V, E^+)$ of a DAG $G = (V, E)$ contains the edge $(u, v) \in E^+$ if and only if there is a directed path in G from u to v. In case G is a tree, the transitive closure can be computed in time $O(|V|^2)$ using Nuutila's algorithm [13]. The MS^n data gives additional information about the provenience of certain peaks/colors, but does not differentiate between different explanations of these peaks via molecular formulas, so we will score not edges but pairs of colors.

To score the closure, let $w^+ : \mathcal{C}^2 \to \mathbb{R}$ be a weighting function for pairs of colors. We define the *transitive weight* of an induced tree $T = (V_T, E_T)$ with transitive closure $T^+ = (V_T, E_T^+)$ as:

$$w^+(T) := \sum\nolimits_{(u,v) \in E_T^+} w^+\big(c(u), c(v)\big) \tag{1}$$

Again, we limit our search to colorful trees, where each color is used at most once in the tree. Scoring the transitive closure of an induced colorful subtree, we reach the following problem definition:

Colorful Subtree Closure problem. Given a vertex-colored DAG $G = (V, E)$ with colors $\mathcal{C}$ and transitive weights $w^+ : \mathcal{C}^2 \to \mathbb{R}$. Find the induced colorful subtree T of G of maximum weight $w^+(T)$.

We will see in the next section that this is again a computationally hard problem. But the problem we are interested in is even harder as it combines the two above problems:

Combined Colorful Subtree problem. Given a vertex-colored DAG $G = (V, E)$ with colors $\mathcal{C}$, edge weights $w : E \to \mathbb{R}$, and transitive weights $w^+ : \mathcal{C}^2 \to \mathbb{R}$. Find the induced colorful subtree T of G of maximum weight $w^*(T) = w(T) + w^+(T)$.

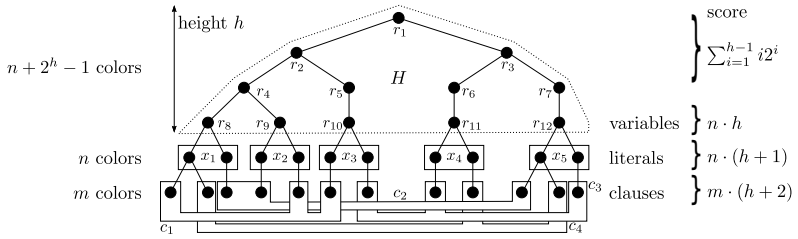

Fig. 1. Proof of Theorem 1: Example for the construction of G for $\Phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_5) \wedge (\overline{x_3} \vee x_4 \vee \overline{x_5}) \wedge (\overline{x_4} \vee x_5 \vee x_1)$.

3 Hardness Results

Fellows *et al.* [8] and Böcker and Rasche [6] independently showed that the Maximum Colorful Subtree problem is NP-hard. It turns out that the Colorful Subtree Closure problem is NP-hard even for unit weights:

Theorem 1. *The* Colorful Subtree Closure *problem is NP-hard even if the input graph is a binary tree with unit weights* $w^+ \equiv 1$.

Proof. To prove NP-hardness we use a reduction from the NP-hard 3-Sat* problem [3]:

3-SAT*. Given a Boolean expression in conjunctive normal form (CNF) consisting of a set of length three clauses, where each variable occurs at most three times in the clause set. Decide whether the expression is satisfiable.

Given an instance of 3-SAT* as a CNF formula $\Phi = c_1 \wedge \cdots \wedge c_m$ over variables $x_1, \ldots, x_n$ we construct an instance of Colorful Subtree Closure. Since variables occurring only with one literal are trivial, we assume that the formula contains both literals of each variable. We first construct a colorful binary tree H with root vertex r and n leaves, that has height $h := \lceil \log_2 n \rceil$ and is a perfect binary tree up to height $h-1$. This tree uses $p := n + 2^h - 1$ colors, namely $r_1, \ldots, r_p$. To each leaf i, $1 \leq i \leq n$, we connect two vertices using the same color x_i and representing the different truth assignments for x_i. One vertex in the color x_i represents $x_i = \mathit{true}$, the other one $x_i = \mathit{false}$. If a truth assignment to x_i satisfies clause c_j we connect a vertex colored c_j to the vertex in the color x_i, that corresponds to this truth assignment (Fig. 1). The resulting tree G possesses $n + 2^h - 1 + n + m$ colors, namely $r_1, \ldots, r_p, x_1, \ldots, x_n, c_1, \ldots, c_m$. The tree is binary, since each variable occurs in at most three clauses and we assumed that both literals are contained in the formula. Finally, we define unit weights $w^+ \equiv 1$.

The resulting tree G has as many leaves as there are literals in Φ, hence the construction is polynomial. We claim that Φ is satisfiable if and only if the colorful subtree T of G with maximum transitive closure has score $\sum_{i=1}^{h-1} i2^i + nh + n(h+1) + m(h+2)$. To prove the forward direction, assume a truth assignment ϕ that satisfies Φ. Define $A \subseteq V(G)$ to be the subset of vertices in the

colors x_i that correspond to the assignment ϕ. Then, for every $1 \leq j \leq m$ there exists at least one vertex colored c_j in the neighborhood of A. Add an arbitrary representative of these vertices colored c_j to the set $B \subseteq V(G)$. The union of the sets $A \cup B \cup \{r_1, \ldots, r_p\}$ forms a colorful subtree T of G with transitive closure that has score $\sum_{i=1}^{h-1} i2^i + nh + n(h+1) + m(h+2)$, as $\sum_{i=1}^{h-1} i2^i$ corresponds to the score of the perfect binary tree up to height $h-1$, nh is the additional score from the leaves of H, $n(h+1)$ the additional score induced by the colors x_i and $m(h+2)$ the additional score induced by the colors c_j. No colorful subtree with higher score of the transitive closure can exist.

To prove the backward direction assume there is a colorful subtree T of G with maximum transitive closure that has score $\sum_{i=1}^{h-1} i2^i + nh + n(h+1) + m(h+2)$. Any optimal solution uses all colors from H, all colors x_i and all colors c_j. The truth assignment corresponding to the vertices of T colored x_i satisfies Φ, as for all $1 \leq j \leq m$ exactly one vertex colored c_j is connected to these vertices, otherwise T would not contain all colors. □

We now turn to the inapproximability of the above problems. Dondi *et al.* [7] show that the Maximum Motif problem, that is closely related to the Maximum Colorful Subtree problem, is APX-hard even if the input graph is a binary tree. In fact, Proposition 8 in [7] implies that the Maximum Colorful Subtree problem is APX-hard for such trees. We infer that there exists no Polynomial Time Approximation Scheme (PTAS) for the problem unless P = NP [1]. In Proposition 10 Dondi *et al.* [7] prove the even stronger result that there is no constant-factor approximation for Maximum Level Motif problem, unless P = NP.

Lemma 1. *The* Maximum Colorful Subtree *problem is APX-hard even if the input graph is a binary tree with unit weights* $w \equiv 1$.

Lemma 2. *The* Maximum Colorful Subtree *problem has no constant-factor approximation unless P = NP, even if the input graph is a tree with unit weights* $w \equiv 1$.

We now concentrate on the Colorful Subtree Closure problem. We show that the problem is MAX SNP-hard even for unit weights, but we have to drop the requirement that the tree is binary in this case. We infer the non-existence of a PTAS unless P = NP [1].

Theorem 2. *The* Colorful Subtree Closure *problem is MAX SNP-hard even if the input graph is a tree with unit weights* $w^+ \equiv 1$.

The construction used in the proof of Theorem 2 is very similar to that of Theorem 1. We defer the details to the full version of the paper.

We infer that the Combined Colorful Subtree problem is computationally hard and also hard to approximate, as it generalizes the above two problems. Note that the input graphs in our application are transitive graphs, whereas we

assume trees in our hardness proofs. One might argue that the problem is actually simpler for transitive graphs; but for a given tree $T = (V, E)$ with unit weights, its transitive closure $G := T^+$ can be complemented with a binary weighting $w : E^+ \to \{0, 1\}$ such that $w(e) = 1$ if and only if $e \in E$. So, the Colorful Subtree Problem remains hard for transitive input graphs. Also note that the input graphs constructed from mass spectra, possess a topological sorting that respects colors. Again, one might argue that the problem is actually simpler for such graphs. It turns out that this is not the case, either: Dondi *et al.* [7] show that the Maximum Motif problem is APX-hard even for leveled trees. All the trees constructed in our reductions are leveled, and all our results are also valid on leveled trees. Similar to above, leveled trees can be encoded in "color-sorted" graphs using binary weightings. Thus, the problems remain hard for graphs with this property.

The Maximum Colorful Subtree problem becomes tractable if the input graph is a colorful graph with non-negative edge weights. But the Colorful Subtree Closure problem remains hard, even in this case:

Theorem 3. *The* Colorful Subtree Closure *problem is MAX SNP-hard even if the input graph is a colorful DAG with a single source and binary weights.*

As we consider a colorful DAG, we can discard all colors and search for a subtree with maximum transitive closure. The transitive closure need not to be defined on colors, but can also be defined on vertices. So, each transitive edge has individual 0/1 weight. We defer the proof of Theorem 3 to the full version of the paper. This proof can be easily adapted to a DAG with maximal vertex degree three.

Surprisingly, we can still find a swift and exact algorithm for the Colorful Subtree Closure problem, presented in the next section.

4 An Exact Algorithm for the Combined Colorful Subtree Problem

Several heuristics for the simpler Maximum Colorful Subtree problem have been evaluated both regarding quality of scores [6] and quality of fragmentation tree [17]. Results of using only the heuristics were of appalling quality, so we refrain from using only heuristics to solve the Combined Colorful Subtree problem. Furthermore, no constant-factor approximation can exist, unless P = NP. But despite the hardness of the problem, we will now present an exact algorithm with reasonable running time in applications. The algorithm is fixed-parameter tractable with respect to the number of colors $k = |C|$, and uses dynamic programming to find the optimum. Note that in application, we can choose k arbitrarily, see below. Let $n := |V|$ and $m := |E|$ be the number of vertices and edges in the input graph $G = (V, E)$, respectively.

Let $W^*(v,S)$ be the maximum score $w^*(T)$ of a colorful subtree with root v using colors $S \subseteq C$. Then $W^*(v,S)$ can be calculated as

$$W^*(v,S) = \max \begin{cases} \max\limits_{u:c(u)\in S\setminus\{c(v)\}} \left\{ \begin{array}{c} W^*(u, S\setminus\{c(v)\}) + w(v,u) \\ + \sum\limits_{c'\in S\setminus\{c(v)\}} w^+\big(c(v),c'\big) \end{array} \right\} \\ \max\limits_{\substack{(S_1,S_2):S_1\cap S_2=\{c(v)\} \\ S_1\cup S_2=S}} W^*(v,S_1) + W^*(v,S_2) \end{cases} \tag{2}$$

where, obviously, we have to exclude the cases $S_1 = \{c(v)\}$ and $S_2 = \{c(v)\}$ from the computation of the second maximum. We initialize $W^*(v,\{c(v)\}) = 0$, and set the weight of nonexistent edges to $-\infty$. To prove the correctness of recurrence (2), we note that we only have to differentiate three cases: A subtree root v can have no children, one child, or two or more children. The case of no children, that is v is a leaf, is covered by the initialization. If v has one child u, we add the score of the tree rooted in u, the score of the new edge (v,u), and scores of all new transitive edges. This is done in the first line of the recurrence. If v has two or more children, we can "glue together" two trees rooted in v, where we arbitrarily distribute the children of v and the colors of S to the two trees.

We now analyze the running time of recurrence (2). Extending a tree by a single vertex takes $O(2^k m)$ time, as we can calculate the sum in constant time by going over the 2^k partitions in a reasonable order. Gluing together two trees, the k colors are partitioned into three groups: those not contained in S, elements of S_1, and elements of S_2. There are 3^k possibilities to perform this partition, so running time is $O(3^k n)$. This results in a total running time of $O(3^k n + 2^k m)$. Running time can be improved to $O^*(2^k)$ using subset convolutions and the Möbius transform [4], but this is of theoretical interest only. In comparison to the algorithm presented in [6], the worst-case running time is not affected by scoring the transitive closure. The necessary space is $O(2^k n)$. In our implementation, we only iterate over defined values in W^*: An entry is not defined if there exists no subtree rooted in v using exactly the colors in S. This algorithm engineering technique does not improve worst-case running times and memory consumption, but greatly improves them in practice. To decrease memory consumption, we use hash maps instead of arrays.

Unfortunately, the above method is limited by its memory and time consumption. In application, exact calculations are limited to $k' \leq k$ colors for some moderate k', such as $k' = 15$. These colors correspond to the k' most intense peaks in the mass spectra, as these contribute most to our scoring. The remaining peaks are added in descending intensity order by a greedy heuristic: For each vertex v with an unassigned color, we try to attach v to every vertex u of the tree constructed so far, where some or all of the children of u in the tree may become children of v. We omit the technical details, and just note that our heuristic is inspired by Kruskal's algorithm for computing a maximum spanning tree.

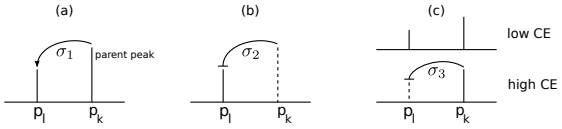

Fig. 2. Scoring of the transitive closure referring to the three cases. A dashed peak is not occurring in the spectrum drawn but in another one (typically the MS^2 spectrum). A connection $p_k \to p_l$ indicates that p_l is a fragment of p_k while a connection $p_k \dashv p_l$ indicates that peak p_l is unlikely to be a fragment of p_k.

5 Scoring

Particularly in fragmentation spectra, the charge of metabolites is mostly ± 1, so we may assume that m/z and mass are equal. Note that our calculations are not limited to a charge of one, though.

The transitive closure score w^+ is defined using the MS^n data. Recall that this score is defined for peaks or, equivalently, colors. In detail, we score three cases, see Fig. 2:

- A spectrum with parent peak p_k and peak p_l indicates that the fragment corresponding to p_l has evolved from the fragment corresponding to p_k. To reward this, we increase the transitive score of the tree by $\sigma_1 \geq 0$ if the fragment corresponding to p_k is a direct or indirect ancestor of the fragment corresponding to p_l, see Fig. 2(a).
- Given a spectrum that contains peak p_l but not peak p_k, and mass $p_l <$ mass p_k. This indicates that the fragment corresponding to p_l has not evolved from the fragment corresponding to p_k. To penalize this, we add $\sigma_2 \leq 0$ to the score if the fragment corresponding to p_k is a direct or indirect ancestor of the fragment corresponding to p_l, see Fig. 2(b).
- Given two spectra with different collision energies and two peaks p_k and p_l with mass $p_l <$ mass p_k. If the spectrum with higher collision energy contains only p_k but the spectrum with lower collision energy contains both peaks, the fragment corresponding to p_l has probably not evolved from the fragment corresponding to p_k. To penalize this case, we add $\sigma_3 \leq 0$ to the score if the fragment corresponding to p_k is a direct or indirect ancestor of the fragment corresponding to p_l, see Fig. 2(c).

In all cases, σ_1, σ_2, and σ_3 are not used to score edges of the fragmentation tree but instead, edges of the transitive closure of the tree. Two peaks are identified to correspond to the same fragment if their masses differ in less than 0.1 Da. For each fragment only the peak with maximum intensity is taken into account for further calculations.

The scoring scheme of the fragmentation graph is the same as introduced in [6], taking the following properties into account: peak intensities, mass deviation between explanation and peak, chemical properties, collision energies and neutral losses. First, every peak is given a base score of b, $b \geq 0$. To score

the mass deviation we evaluate the logarithmized Gaussian probability density function with SD σ at the measuring error value. Further we use the density function of the normal distribution with mean 0.59 and SD 0.56 to score the hetero atom to carbon ratio of the decompositions. Due to the collision energies of the different spectra, some peaks cannot represent fragments of other peaks. A fragment appearing at lower collision energy than its predecessor is penalized with $\log(\alpha)$, $\alpha \ll 1$. If there is no spectrum containing both, neither containing none of the peaks we add only a penalty of $\log(\beta)$, $\alpha < \beta < 1$. Common neutral losses are rewarded with $\log(\gamma)$, $\gamma > 1$, while radical neutral losses are penalized by $\log(\delta)$, $\delta < 1$, and large neutral losses by $\log(1 - \frac{\text{mass(neutral loss)}}{\text{parent mass}})$. In addition to the scoring from [6], we use an extension that takes into account *rare* neutral losses: If a rare neutral loss occurs in a fragmentation step we penalize it by adding $\log(\eta), \eta \ll 1$. We also penalize neutral losses that consists carbon or only nitrogen atoms by adding $\log(\epsilon), \epsilon \ll 1$. In contrast, radical losses are not penalized, since they sometimes occur in fragmentation reactions. Due to space constraints, we defer a list of all rare neutral losses and radical losses to the full version of the paper.

6 Results

To evaluate our work we implemented the algorithm in Java 1.6. As test data we used 185 mass spectra of 45 compounds, mostly representing plant secondary metabolites. The 185 mass spectra are composed of 45 MS^2 spectra, 128 MS^3 spectra and twelve MS^4 spectra (unpublished). All spectra were measured on a Thermo Scientific Orbitrap XL instrument, we omit the experimental details. Peak picking was performed using the Xcalibur software supplied with the instrument. The data set was analyzed with the following options: For decomposing peak masses we use a relative error of 20 ppm and the standard alphabet containing carbon (C), hydrogen (H), nitrogen (N), oxygen (O), phosphorus (P), and sulfur (S). For the construction of the fragmentation graph, we use the collision energy scoring parameters $\alpha = 0.1$, $\beta = 0.8$, the neutral loss scoring parameters $\gamma = 10$, $\delta = 10^{-3}$, $\epsilon = 10^{-4}$, $\eta = 10^{-3}$, the intensity scoring parameter $\lambda = 0.1$, the base score $b = 0$ and the standard deviation of the mass error $\sigma = 20/3$ as described in [6, 17]. We can identify the molecular formulas of the compounds from isotope pattern analysis and by calculating the fragmentation trees for all candidate molecular formulas [17]. In this paper, we assume that this task has been solved beforehand, and that all molecular formulas are known.

Comparing Trees. We evaluate the impact of using MS^n instead of MS^2 data, as well as the influence of scoring parameters σ_1, σ_2, σ_3 from Sec. 5, using pairwise tree comparison. In each fragmentation tree, vertices are implicitly labeled by molecular formulas of the corresponding fragments. We limit our comparison to those fragments that appear in both trees, and discard orphan fragments. We distinguish four cases:

- A fragment is *identically placed*, if its parent fragments are identical in both trees.
- A fragment is *pulled up*, if its parent fragment in the second tree is one of its predecessors in the first tree (and the fragment is not identically placed).
- A fragment is *pulled down*, if its parent fragment in the first tree is one of its predecessors in the second tree (and the fragment is not identically placed).
- A fragment is *regrafted*, if it is not identically placed, pulled up or pulled down.

The obvious way to evaluate our method would be to compare our results against some gold standard. Unfortunately, such gold standard is not available for our study. Rasche *et al.* [17] have evaluated the method from [6] by expert annotation of MS^2 fragmentation trees for a subset of the compounds used in this paper. Unfortunately, the input data (that is, fragments observed in the MS^2 and MS^n mode of the instrument) differ strongly. Hence, a comparison against these expert-annotated fragmentation trees is impossible.

As mentioned in Sec. 4 the exact algorithm is memory and time consuming. So, we use the exact algorithm for only the k' most intense peaks, and attach the remaining peaks using a greedy heuristic. We find that decreasing k' from 20 to 15, has a comparatively small effect on the computed fragmentation trees: 97.1% of the fragments were identically placed, 0.4% were pulled up, 0.6% were pulled down, and only 1.9% were regrafted. On the other hand, average running time per compound was decreased from 30.8 min to 3.97 s. In the remainder of this section, we set $k' = 15$ and use only the 15 most intensive peaks for exact computations. Choosing a moderate $k' = 15$ has a much stronger effect here, than it was observed for the Maximum Colorful Subtree problem [6], where constructed fragmentation trees were practically identical for $k' = 15$ and $k' = 20$. We attribute this to the transitive scoring, which appears to be harder to grasp by the heuristic.

To show the effect of evaluating MS^n data, we individually varied the three score parameters, and compared the resulting trees to the trees constructed

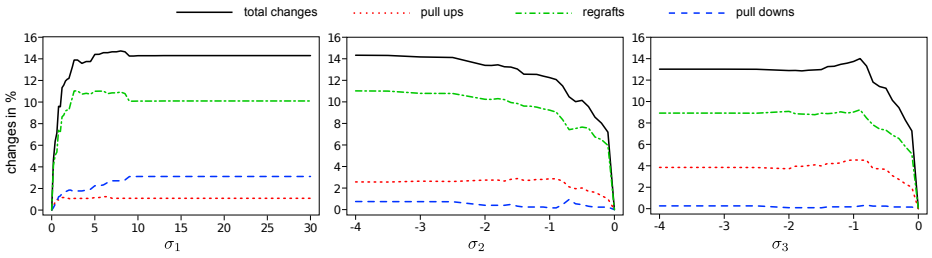

Fig. 3. Percentage of pull ups, pull downs, regrafted fragments, and total changed fragments when varying score parameters σ_1, σ_2, and σ_3. Left: Varying σ_1 with $\sigma_2 = 0$ and $\sigma_2 = 0$ fixed. Middle: Varying σ_2 with $\sigma_1 = 0$ and $\sigma_3 = 0$ fixed. Right: Varying σ_3 with $\sigma_1 = 0$ and $\sigma_2 = 0$ fixed.

without scoring the transitive closure, see Fig. 3. As σ_1 increases, the fraction of changes in the trees (pull ups, pull downs and regrafts) converges to about 14%. A similar behavior is observed as σ_2, σ_3 are decreased. The main difference between the bonus score σ_1 and the penalty scores σ_2 and σ_3 is that increasing σ_1 results in more pull downs than pull ups, while decreasing penalty scores σ_2, σ_3 produces more pull ups than pull downs. This can be explained as follows: Reward scores can rather be realized if fragments are inserted deep, that is, far from the root. In contrast, negative penalty scores are avoided if the fragments are inserted "shallow", that is, close to the root. So, $\sigma_1 \gg 0$ tends to deepen the trees, whereas $\sigma_2, \sigma_3 \ll 0$ tends to broaden the trees.

Based on the above analysis, we decided to use the following parameter values: $k' = 15$, $\sigma_1 = 3$, $\sigma_2 = -0.5$, and $\sigma_3 = -0.5$. We choose a large σ_1 as the underlying MS^n observation is a clear signal that some fragment should be placed as a successor of another fragment. In comparison, the MS^n reasoning behind σ_2 and σ_3 is somewhat weaker, so we choose smaller absolute values for these parameters that less influence the trees. The crucial comparison is now between the fragmentation trees computed without scoring the transitive closure and the fragmentation trees computed with the above scores. As we have only one MS^2 spectrum per compound and one spectrum contains too few peaks to calculate a reasonable tree, we transform the MS^n data to "pseudo-MS^2" data by merging all fragmentation spectra of a compound into one. This simulates MS^2 spectra with different collision energies. By merging all spectra into one, we loose all information about dependencies between peaks/colors. This is implicitly achieved by setting $\sigma_1, \sigma_2, \sigma_3 = 0$. Between these trees 76.21% of the fragments are identically connected, 4.90% are pull ups, 1.79% pull downs and 17.11% regrafted fragments. Hence, almost one quarter of all fragments are changed due to the information from MS^n data.

We cannot evaluate whether these changed neutral losses are true or false and, hence, whether MS^n fragmentation trees are truly better than the MS^2 trees. But we will now show an example where the MS^n tree agrees well with the observed MS^n data: To this end, we consider the fragmentation trees of Phenylalanine, with and without scoring the transitive closure, see Fig. 4. The two fragmentation trees are almost identical, with the single exception of fragment C_7H_9 at 93.1 Da: This fragment is connected to $C_9H_9O_2$ at 149.0 Da in the MS^2 tree, and to $C_8H_{10}N$ at 120.1 Da in the MS^n tree. In the MS^2 interpretation, the neutral loss C_2O_2 is explained as two common neutral losses CO and, hence, it is preferred over the neutral loss CHN (hydrogen cyanide). Using MS^n data, we can resolve this: the peak at 93.1 Da *does* occur in the MS^3 spectrum with parent peak at 120.1 Da, therefore C_7H_9 at 93.1 Da probably resulted (directly or indirectly) as a fragment of $C_8H_{10}N$ at 120.1 Da. This is rewarded by our algorithm, adding $\sigma_1 = +3$ to the score of the modified tree. The fact that the peak at 107.0 Da is missing in the MS^3 spectrum with parent peak at 120.1 Da, does not change the score: In the MS^2 analysis, fragment C_7H_7O cannot be a successor of $C_8H_{10}N$ at 120.1 Da, nor are 91.1 Da, 93.1 Da, or 103.1 Da assumed to be its successor. Another example where the MS^n tree agrees well with the observed MS^n data

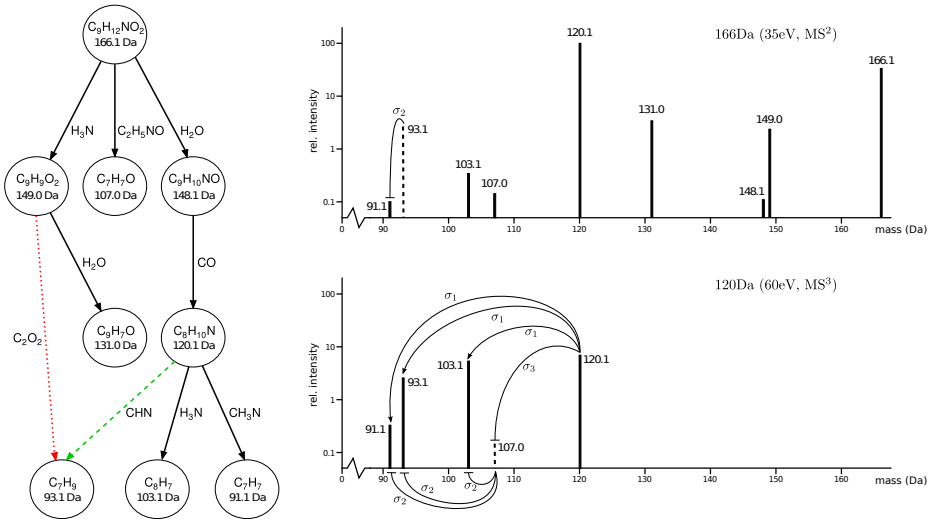

Fig. 4. Left: Fragmentation trees of phenylalanine. Solid edges are neutral losses present in both trees, the red dotted (green dashed) edge is present in the MS^2 (or MS^n) tree only, respectively. Right: MS^2 spectrum of the parent peak (top) and MS^3 spectrum of the 120.1 Da fragment (bottom). Dashed peaks are not contained in the particular spectrum.

is tryptophan. Due to space constraints, we defer the details of this analysis to the full version of the paper.

As shown in Sec. 3, the theoretical worst case running time of our algorithm is identical with that of the Maximum Colorful Subtree algorithm in [6]. We investigated whether this also holds in application. Running times were measured on an Intel Core 2 Duo, 2.4 GHz with 4 GB memory, with parameter $k' = 15$. We find that total running times of the algorithm, with and without using MS^n data, are practically identical: Average running time is about 3.8 s, and the maximal running time for one compound was 17.6 s. We omit further details.

7 Conclusion

In this paper, we have presented a framework for computing metabolite fragmentation trees using MS^n data. Our fragmentation model results in the Combined Colorful Subtree problem, a conjunction of the Maximum Colorful Subtree problem and the Colorful Subtree Closure problem. Both problems are NP-hard, and no PTAS can exist for either problem. The latter problem remains MAX SNP-hard even if the input graph is colorful.

We have presented an exact dynamic programming algorithm for the Combined Colorful Subtree problem, showing that the problem is fixed-parameter tractable with respect to the parameter "number of colors". We have intro-

duced a scoring scheme based on the dependencies between the different fragmentation steps. To reduce memory and time requirements, we limit exact computations to the $k' \leq k$ most intense peaks in the spectrum. Although the Combined Colorful Subtree problem is computationally hard, the resulting algorithm is fast in practice.

For our application, the score of the transitive closure $w^+ : \mathcal{C}^2 \to \mathbb{R}$ is defined on *pairs of colors*. From the theoretical standpoint, one can modify the problem such that the score $w^+ : E^+ \to \mathbb{R}$ is defined on *edges of the transitive closure* of the fragmentation graph $G = (V, E)$. In this case, our algorithm from Sec. 4 cannot be used, and it remains an open problem whether this modified version of the Colorful Subtree Closure problem is fixed-parameter tractable with respect to the number of colors. Clearly, the problem is in FPT for unit weights.

We have seen that using additional information from MS^n data does change the computed fragmentation trees. In our experiments, one quarter of fragments were differently inserted when including MS^n information. As our scoring scheme is "chemically reasonable", we argue that the trees are actually improved using MS^n data. Unfortunately, MS^n is less suited for high-throughput measurements, as individual measurements are more time-consuming. On the other hand, for about three quarters of the fragments, trees remain identical between MS^2 and MS^n. Thus, calculating fragmentation trees from MS^2 data extracts valuable information concealed in these spectra and results in largely reasonable trees.

In the future, we want to increase the speed and decrease the memory consumption of our exact algorithm. Also, we want to use MS^n fragmentation trees to fine-tune the scoring parameters for computing MS^2 fragmentation trees. The next step of the analysis pipeline is a method for automated comparison of fragmentation trees.

Acknowledgments. We thank Aleš Svatoš and Ravi Kumar Maddula from the Max Planck Institute for Chemical Ecology in Jena, Germany for supplying us with the test data. K. Scheubert was funded by Deutsche Forschungsgemeinschaft, project "IDUN". F. Hufsky was supported by the International Max Planck Research School Jena.

References

1. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.
2. N. Bandeira, J. V. Olsen, J. V. Mann, M. Mann, and P. A. Pevzner. Multi-spectra peptide sequencing and its applications to multistage mass spectrometry. *Bioinformatics*, 24(13):i416–i423, Jul 2008.
3. P. Berman, M. Karpinski, and A. D. Scott. Computational complexity of some restricted instances of 3-SAT. *Discrete Appl. Math.*, 155:649–653, 2007.
4. A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. of ACM Symposium on Theory of Computing (STOC 2007)*, pages 67–74. ACM Press New York, 2007.
5. S. Böcker and Zs. Lipták. A fast and simple algorithm for the Money Changing Problem. *Algorithmica*, 48(4):413–432, 2007.

6. S. Böcker and F. Rasche. 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).
7. R. Dondi, G. Fertin, and S. Vialette. Complexity issues in vertex-colored graph pattern matching. *J. Discrete Algorithms*, 2010. In press, doi:10.1016/j.jda.2010.09.002.
8. M. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP 2007)*, volume 4596 of *Lect. Notes Comput. Sc.*, pages 340–351. Springer, 2007.
9. A. R. Fernie, R. N. Trethewey, A. J. Krotzky, and L. Willmitzer. Metabolite profiling: from diagnostics to systems biology. *Nat. Rev. Mol. Cell Biol.*, 5(9):763–769, 2004.
10. M. Heinonen, A. Rantanen, T. Mielikäinen, J. Kokkonen, J. Kiuru, R. A. Ketola, and J. Rousu. 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.
11. D. W. Hill, T. M. Kertesz, D. Fontaine, R. Friedman, and D. F. Grant. Mass spectral metabonomics beyond elemental formula: Chemical database querying by matching experimental with computational fragmentation spectra. *Anal. Chem.*, 80(14):5574–5582, 2008.
12. F. W. McLafferty and F. Tureček. *Interpretation of Mass Spectra*. University Science Books, Mill valley, California, fourth edition, 1993.
13. E. Nuutila. An efficient transitive closure algorithm for cyclic digraphs. *Inform. Process. Lett.*, 52(4):207–213, 1994.
14. H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and inter-laboratory transferability of a tandem mass spectral reference library: 1. results of an austrian multicenter study. *J. Mass Spectrom.*, 44(4):485–493, 2009.
15. H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and the inter-laboratory transferability of a tandem mass spectral reference library: 2. optimization and characterization of the search algorithm. *J. Mass Spectrom.*, 44(4):494–502, 2009.
16. A. Pelander, E. Tyrkkö, and I. Ojanperä. In silico methods for predicting metabolism and mass fragmentation applied to quetiapine in liquid chromatography/time-of-flight mass spectrometry urine drug screening. *Rapid Commun. Mass Spectrom.*, 23(4):506–514, 2009.
17. F. Rasche, A. Svatoš, R. K. Maddula, C. Böttcher, and S. Böcker. Computing fragmentation trees from tandem mass spectrometry data. *Anal. Chem.*, Dec 2010. In press, doi:10.1021/ac101825k.
18. M. T. Sheldon, R. Mistrik, and T. R. Croley. Determination of ion structures in structurally related compounds using precursor ion fingerprinting. *J. Am. Soc. Mass Spectrom.*, 20(3):370–376, 2009.