Universität Bielefeld

Technische Fakultät
Abteilung Informationstechnik
Forschungsberichte

Identifying metabolites with integer decomposition techniques, using only their mass spectrometric isotope patterns

Sebastian Böcker
Matthias C. Letzel
Zsuzsanna Lipták
Anton Pervukhin

Report 2007-01

Impressum: Herausgeber:
Ellen Baake, Robert Giegerich, Ralf Hofestädt, Franz Kummert,
Peter Ladkin, Ralf Möller, Helge Ritter, Gerhard Sagerer,
Jens Stoye, Holger Theisel, Ipke Wachsmuth

Technische Fakultät der Universität Bielefeld,
Abteilung Informationstechnik, Postfach 10 01 31,
33501 Bielefeld, Germany

ISSN 0946-7831

Identifying metabolites with integer decomposition techniques, using only their mass spectrometric isotope patterns

SEBASTIAN BÖCKER[1] and MATTHIAS C. LETZEL[2] and
ZSUZSANNA LIPTÁK[3] and ANTON PERVUKHIN[1]

[1] Lehrstuhl für Bioinformatik, Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2, 07743 Jena, Germany, boecker,apervukh@minet.uni-jena.de
[2] Organische Chemie I, Massenspektrometrie, Fakultät für Chemie
matthias.letzel@uni-bielefeld.de
[3] AG Genominformatik, Technische Fakultät zsuzsa@CeBiTec.uni-bielefeld.de
Universität Bielefeld, PF 100 131, 33501 Bielefeld, Germany

Abstract. [4] Metabolites, small molecules that are intermediates and products of the metabolism, participate in almost all cellular processes such as signal transduction and stress response. There exist several thousand metabolites for every species, the overwhelming majority still being uncharacterized. Mass spectrometry has become a method of choice to analyze the metabolites of a cell. High resolution mass spectrometry allows us to determine the mass and isotopic distribution of sample molecules with outstanding accuracy. Here, we provide a method to determine the sum formula of an unidentified metabolite (or, more generally, any chemical compound) solely from its mass and isotopic pattern. This is a crucial step in the identification of an unknown metabolite, as it reduces its possible structures to a finite and, hopefully, manageable set.

In Part I, we show how to use integer decomposition techniques, introduced earlier by two of the authors, for decomposing real valued molecule masses, with large improvements over naïve methods that are currently best known for this problem. We then show how to rapidly match and rank simulated spectra against the measured spectrum. Our method is computationally efficient and can be applied to metabolites and other chemical compounds with mass up to 1000 Dalton. First results on experimental data indicate good identification rates for chemical compounds up to 700 Dalton.

In Part II, we present our method for rapid computation of isotope distributions and mean masses of isotope peaks, i.e., for simulation of isotopic spectra, improving on best-known results. Fast simulation of isotope patterns is vital due to the large search space. Above 1000 Dalton, however, the number of molecules with a certain mass increases rapidly. Since the size of the search space thus becomes prohibitive, generating all

[4] A shorter version of this paper appeared in the proceedings of the 6th Workshop on Algorithms in Bioinformatics (WABI 2006), volume 4175 of LNBI/LNCS, pages 12-23, Springer 2006.

potential solutions, simulating their isotope patterns, and matching them against the input is often not feasible. Instead, we define several *additive invariants* extracted from the input and then propose to solve a *joint decomposition problem*: Given a finite weighted alphabet with character masses $\{a_1, \dots, a_\sigma\}$ and a query m, a *decomposition* of m is a non-negative integer vector $(c_1, \dots, c_\sigma)$ such that $\sum_i c_i a_i = m$. Here, we have the problem of finding a *joint* decomposition c for a set of queries, where each query has to be decomposed over a different weighted alphabet. We present an efficient algorithm for producing all joint decompositions of the query vector and demonstrate its fitness on real data extracted from a metabolite database.

Part I

Identifying metabolites using high precision mass spectrometry

1 Introduction to Part I

The term "metabolite" is usually restricted to small molecules that are intermediates and products of the metabolism. These small molecules participate in almost all cellular processes such as signal transduction, stress response, catabolism, or anabolism. It is widely accepted that every species hosts several thousand metabolites; however, the overwhelming majority of these metabolites is yet uncharacterized. The majority of metabolites have mass below 1000 Dalton: 96.5 % of sum formulas in the KEGG LIGAND database fall into this mass range [9].

Mass spectrometry, along with nuclear magnetic resonance spectroscopy, has become the method of choice to analyze the metabolites of a cell. Today, metabolites are usually identified through fragmenting the metabolite using electron impact ionization, and subsequent database lookup in a chemical compound library [15]. Clearly, this method is limited to identifying metabolites and chemical compounds that have been included in some library.

High resolution mass spectrometry, such as Fourier Transform Ion Cyclotron Resonance mass spectrometry, allows us to determine the mass of a sample molecule with an accuracy of about one thousandth of a single proton mass. Using the mass and the isotopic pattern of an unknown metabolite, one can try to identify the sum formula of the metabolite, that is, the number of atoms of each element that make up the individual molecule. This is a crucial step in identifying the unknown metabolite, because a fixed sum formula reduces the number of possible structures to a closed set that can be further evaluated by approaches for automatic structure elucidation. In the following, when talking about "identifying a molecule" we refer to determination of its sum formula.

Molecules in the sample are separated using, say, liquid chromatography and inserted into the mass spectrometer. After preprocessing, the output of a mass spectrometry experiment is a list of peaks which ideally correspond to masses and relative abundances of sample molecules and their isotopes. If a mixture of molecules is present, then separating peaks that belong to different molecules is a trivial task except for the very rare cases where peaks "overlap." For readability, we assume that our input is a vector of peak masses $M_0, \dots, M_K$ and intensities $f_0, \dots, f_K$ corresponding to the isotopic distribution of a single molecule.

A straightforward approach of using this information for the molecule's identification is to generate all molecules with monoisotopic mass sufficiently close to M_0, compute the isotopic distribution of the candidate molecules, and compare these simulated distributions to the measured data. [11] investigate the resolving power of isotopic distributions using simulations, but ignore mean peak masses. In 2006, [7] and [16] used high-precision mass spectrometry to infer sum formulas of unknown molecules with mass below 321 Dalton. To the best of our knowledge, these are the first studies reported in literature where sum formulas are derived solely from molecules' isotope patterns. Both studies focus on the experimental side of the problem. [16] do not give any computational methods, while [7] give only basic computational methods for the automated analysis of isotopic patterns.

The problem of finding all molecules that have monoisotopic mass M_0, has been addressed frequently from the biochemical and mass spectrometry viewpoint [6], but no efficient algorithms for this problem were given. There exist time and space efficient methods to decompose integer masses [4, 5]. In Sec. 3 we use these techniques for the decomposition of real-valued masses.

The number of molecules with mass M_0 increases significantly for large M_0. Thus, the sheer size of the search space makes it necessary to develop efficient methods for simulating the isotopic distribution of a molecule (see Part II) but also to rank candidate molecules with respect to the measured spectrum, see Sec. 4. This initial ranking is rather intended as a filter to efficiently discard candidate molecules that show low agreement with the measured spectrum. As a proof of concept, we have applied our method to high resolution mass spectra.

2 Physical and chemical background

The elements most abundant in living beings are hydrogen (symbol H) with atomic number (i.e., number of protons) 1, carbon (C, atomic number 6), nitrogen (N, 7), oxygen (O, 8), phosphor (P, 15), and sulfur (S, 16). For ease of exposition, we will restrict ourselves to these elements for the remainder of this paper, sometimes even ignoring sulfur; see Section 6 for a generalization to arbitrary elements.

The *mass number* of an atom is its total number of protons and neutrons. Elements can have atoms with equal atomic number but varying number of neutrons, called *isotopes*. Several isotopes of each element can be found in nature: Regarding the elements most abundant in living beings, see Table 1 for all natural isotopes and their relative abundance.

The *mass* of an atom is measured in *unified atomic mass units* with symbol "u" or, equivalently, in "Dalton" (Da). One Da equals 1/12 of the mass of one atom of the ^{12}C isotope, approximately $1.66 \cdot 10^{-27}$ kg. An atom that contains n protons and neutrons will have a mass approximately equal to n Da. This approximation does not account for the mass contained in the binding energy of the atom's nucleus. This explains the *mass defect*, the difference between the atom's mass and the larger sum of masses of the protons, neutrons, and electrons contained: For example, 6 protons, 6 neutrons, and 6 electrons have a total mass of 12.09596 Da while the ^{12}C isotope has a mass of exactly 12.0 Da, a deviation of about 0.8 %. See Table 1 or [1] for a detailed list.

A molecule consists of a stable system of two or more atoms. The *sum formula* describes the number of atoms of the different elements that compose the molecule. The *nominal mass* (also called *nucleon number*) of a molecule is the sum of protons and neutrons of the constituting atoms. The *mass* of a molecule is the sum of masses of the atoms it is composed of. The mass and nominal mass of a molecule depend on the isotopes that constitute it. To this end, the *monoisotopic* (*nominal*) *mass* of a molecule is the sum of (nominal) masses of the constituting atoms where for every element, we choose the natural isotope with smallest mass number. In this paper, the term "monoisotopic" consistently refers to the

element (symbol)	isotope	mass	mass diff.	abundance	av. mass
hydrogen (H)	^{1}H	1.007825		99.985 %	
	^{2}H	2.014102	+1.006277	0.015 %	1.007975
carbon (C)	^{12}C	12.0		98.890 %	
	^{13}C	13.003355	+1.003355	1.110 %	12.011137
nitrogen (N)	^{14}N	14.003074		99.634 %	
	^{15}N	15.000109	+0.997035	0.366 %	14.006727
oxygen (O)	^{16}O	15.994915		99.762 %	
	^{17}O	16.999132	+1.004217	0.038 %	
	^{18}O	17.999161	+2.004246	0.200 %	15.999305
phosphor (P)	^{31}P	30.973762		100 %	30.973762
sulfur (S)	^{32}S	31.972071		95.020 %	
	^{33}S	32.971459	+0.999388	0.750 %	
	^{34}S	33.967867	+1.995796	4.210 %	
	^{36}S	35.967081	+3.995010	0.020 %	32.064388

proton (p^+, $^{1}H^+$) 1.00728 Da, neutron (n) 1.008665 Da, electron (e^-) 0.00054 Da

Table 1. Natural isotopic distribution: Relative abundance of isotopes and their masses in Dalton, rounded to six decimal places.

lightest isotope, not the most abundant isotope. For example, 506.99575 Da is the monoisotopic mass of adenosine triphosphate (ATP) $C_{10}H_{16}N_5O_{13}P_3$ with monoisotopic nominal mass 507.

2.1 Isotope species

Mass spectrometry cannot detect single molecules but is dependent on the existence of millions of identical copies of some molecule.[5] In living beings, this means that elements follow their natural isotopic distribution and instead of identical copies, we have different *isotope species* of a molecule. See Table 2 for isotope species and their relative abundances of ATP.

Given the isotope species of two molecules, we can easily calculate the isotope species of the joined molecule by folding the species (species with masses m_1, m_2 and probabilities p_1, p_2 result in an isotope subspecies with mass $m_1 + m_2$ and probability $p_1 p_2$ in the joined molecule), then sorting the subspecies with respect to mass, and finally merging isotope subspecies with identical mass. The number of isotope species is rather large for medium size molecules, even if we ignore isotope species that show negligible relative abundance (see Part II, Section 9 for details): For example, ATP has 117 810 isotope species. Furthermore, we usually cannot resolve isotope species with identical nominal mass using present-day analysis techniques. Using FT-ICR this is not so much a problem of

[5] More precisely, mass spectrometry cannot detect molecules but ions, molecules that have picked up a net electric charge, while by definition, molecules have no net electric charge. In particular, we have to shift masses according to the appended ion. We ignore this for ease of exposition.

^{12}C	^{13}C	^{1}H	^{2}H	^{14}N	^{15}N	^{16}O	^{17}O	^{18}O	^{31}P	nominal	mass (Da)	abund. %
10	0	16	0	5	0	13	0	0	3	507	506.995751	84.9310
10	0	16	0	4	1	13	0	0	3	508	507.992786	1.5599
9	1	16	0	5	0	13	0	0	3	508	507.999106	9.5331
10	0	16	0	5	0	12	1	0	3	508	507.999968	0.4205
10	0	15	1	5	0	13	0	0	3	508	508.002028	0.2038
10	0	16	0	3	2	13	0	0	3	509	508.989821	0.0114
9	1	16	0	4	1	13	0	0	3	509	508.996141	0.1750
10	0	16	0	4	1	12	1	0	3	509	508.997003	0.0077
10	0	15	1	4	1	13	0	0	3	509	508.999063	0.0037
10	0	16	0	5	0	12	0	1	3	509	508.999997	2.2134
8	2	16	0	5	0	13	0	0	3	509	509.002461	0.4815
9	1	16	0	5	0	12	1	0	3	509	509.003323	0.0472
10	0	16	0	5	0	11	2	0	3	509	509.004185	0.0010
9	1	15	1	5	0	13	0	0	3	509	509.005383	0.0228
10	0	15	1	5	0	12	1	0	3	509	509.006245	0.0010
10	0	14	2	5	0	13	0	0	3	509	509.008305	0.0002

Table 2. Isotope species of adenosine triphosphate (ATP) molecules $C_{10}H_{16}N_5O_{13}P_3$, sorted by mass. Isotope species with nominal mass ≥ 510 omitted.

limited resolution of the mass spectrometer, but of the limited dynamic range of the technique. See Section 6 for possible exceptions such as sulfur-containing molecules.

2.2 Isotopic distributions and mean peak masses

One can simplify matters by combining isotope species with identical nominal mass. Formally, we can represent the distribution of an element E by a discrete random variable Y_E with finite state space $\Omega_E \subseteq \mathbb{N}$: For example, carbon has state space $\Omega_E := \{12, 13\}$ and random variable Y_C with $\mathbb{P}(Y_\mathrm{C} = 12) = 0.98890$ and $\mathbb{P}(Y_\mathrm{C} = 13) = 0.01110$. The resulting distribution of nominal masses is called the *isotopic distribution* of the molecule. In an ideal mass spectrum, normalized peak intensities correspond to these probabilities. We refer to the peak at monoisotopic mass as monoisotopic, and to the following peaks as +1, +2, ... peaks. See Table 3 on page 9 for the isotopic distribution of ATP.

Note that isotope species with distinct nominal masses may have almost identical *real* masses, rendering it impossible to merge isotope species into an isotopic distribution. But if we limit ourselves to the first, say, ten isotope peaks, we can safely assume that such merging is possible: For *every* molecule over the elements CHNOPS, the +10 peak is found between plus 9.97898 Da and plus 10.06277 Da.

Following Part II, Section 10, we can compute the isotopic distribution of an arbitrary molecule as follows: We can restrict ourselves to computing the first K non-zero values of the distribution, for rather small K such as $K = 10$. The isotopic distribution of a molecule E_l consisting of l atoms of element $E \in \{\mathrm{H}, \mathrm{C}, \mathrm{N}\}$ follows a binomial distribution, and can be computed in time $O(K + \log l)$. For other elements, we do not compute distributions on the fly but during preprocessing, for all $l \leq L$ fixed. This results in $O(KL)$ memory for every such element, where L is small in applications: 64 oxygen atoms already have mass of about 1024 Da, exceeding the relevant mass range.

Given two molecules with known isotopic distributions we can compute the distribution of the joined molecule by folding distributions, which requires time $O(K^2)$. So, to find the isotopic distribution of an arbitrary molecule, we fold the distributions of the individual elements that are either present in memory (O, S) or can be computed efficiently (C, H, N). We need $O(nK^2)$ time for n elements.

The imperfection of mass spectrometry results in a +1, +2, ... isotope peak that, in fact, are superpositions of peaks with almost identical mass. What is the mass of such a superposition peak? It is reasonable to assume that its mass is the mean mass of all isotope species that add to its intensity: Given a fixed nominal mass we sum up the masses of all isotopic species of this nominal mass, weighted by their relative abundance.

nominal mass	507	508 (+1)	509 (+2)	510 (+3)	511 (+4)	512 (+5)
abundance %	84.9309	11.7175	2.9653	0.3343	0.0469	0.0044
mean peak m.	506.995751	507.998347	509.000220	510.002655	511.004629	512.006961

Table 3. Mean peak masses and abundances of ATP $C_{10}H_{16}N_5O_{13}P_3$ distribution. Peaks with nominal mass 513 and above have abundances $< 0.001\,\%$.

In Figure 1, we plot the isotope species and mean peaks of ATP.

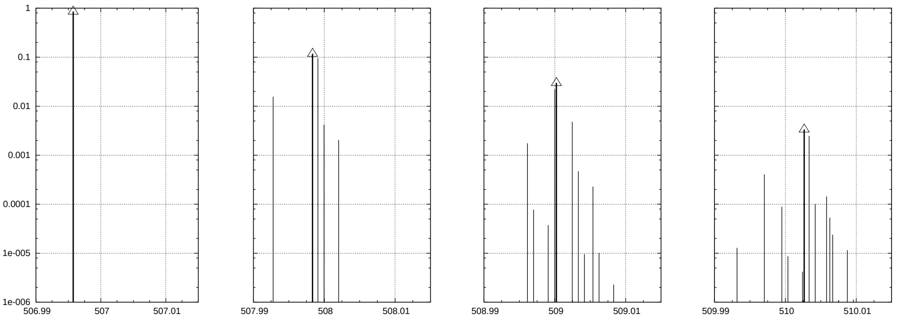

Fig. 1. Isotope species and isotope mean peaks of adenosine triphosphate (ATP) molecules $C_{10}H_{16}N_5O_{13}P_3$. Mean peaks marked with a triangle.

We can compute these masses by *folding* mean peak masses (for details, see Part II, Section 10) analogous to the folding of distributions: We are given two molecules with known isotopic distributions p_k and q_k and known mean peak masses m_k and m'_k, $k \leq K$. Now, the mean peak mass of the $+k$ peak of the joined molecule is:

$$\tilde{m}_k = \tfrac{1}{\tilde{p}_k} \cdot \sum\nolimits_{j=0}^{k} p_j q_{k-j} \left(m_j + m'_{k-j} \right)$$

In the following, the *isotopic pattern* of a molecule is its isotopic distribution plus mean peak masses.

3 Decompositions of real valued numbers

We want to find all molecules with (monoisotopic) mass in the interval $[l, u] \subseteq \mathbb{R}$ where $l := M_0 - \varepsilon$ and $u := M_0 + \varepsilon$ for some measurement inaccuracy ε. Formally, we search for all solutions of the integer knapsack equation [10]

$$a_1 c_1 + a_2 c_2 + \cdots + a_n c_n \in [l, u] \tag{1}$$

where a_j are real-valued monoisotopic masses of elements satisfying $a_j \geq 0$. We search for all solution vectors $c = (c_1, \ldots, c_n)$ such that all c_j are non-negative integers. We may assume $a_1 < a_2 < \cdots < a_n$.

A straight-forward solution is to generate all vectors c with $c_1 = 0$ and $\sum_j a_j c_j \leq u$, and next to test if there is some $c_1 \geq 0$ such that $\sum_j a_j c_j \in [l, u]$. This results in $O(m^{n-1})$ runtime where $m := M_0/a_2$. Alternatively, we can compute all potential decompositions up to some upper bound U during preprocessing, sort them with respect to mass and use binary search; this results in $O(U^n)$ space requirement. These approaches are unfavorable in theoretical complexity as well as in practice: For the alphabet CHNOPS there exist more than $7 \cdot 10^8$ sum formulas with mass below 1000 Da.

In case of integer coefficients, one can use dynamic programming to compute all solutions efficiently, following the line of thought of [10, Sec. 8.3]. In a preprocessing step, a bit table of size $n \times U$ is computed in time $O(nU)$, where $U \in \mathbb{N}$ is the maximal upper bound we want to consider in the following. Using this table, we can efficiently find all solutions (1) for all queries $l, u \leq U$. The main disadvantage of this approach is the memory requirement of $O(nU)$. An alternative method for finding all solutions is given in [4], using a table of size $O(k\, a_1)$. Every solution is constructed in time $O(n a_1)$ independent of the input l, u. In addition, we do not have to choose a maximal bound U we want to consider. Regarding the application of decomposing molecule masses, the latter approach uses only 1/15 of memory and shows slightly better runtimes.

Reconsider the original integer knapsack problem with real-valued coefficients. Choosing a *blowup factor* $b \in \mathbb{R}$, corresponding to precision $1/b$, we can round coefficients by $\varphi(a) := \lceil ba \rceil$, so $a'_j := \varphi(a_j)$ and $l' := \varphi(l)$, $u' := \varphi(u)$ form a Diophantine equation. We stress that precision $1/b$ is merely a parameter of the decomposition algorithm and in principle independent of the measurement mass accuracy ε. To avoid rounding error accumulation, precision is usually set one to two orders of magnitude smaller than the measurement accuracy. Now, certain solutions c of the integer coefficient knapsack are no solutions of the real-valued coefficient knapsack, and vice versa. We can easily sort out false positive solutions checking (1), resulting in additional runtime. But first, we concentrate on the more intriguing problem of false negative solutions that are missed by the integer coefficient knapsack.

Clearly $\sum_j a_j c_j \geq l$ implies $\sum_j a'_j c_j \geq l'$ since all a'_j are integer. We have to increase the upper bound u' to guarantee that all solutions of (1) are generated. We define relative rounding errors

$$\Delta_j = \Delta_j(b) := \frac{\lceil ba_j \rceil - ba_j}{a_j} \quad \text{for } j = 1, \ldots, n$$

where $0 \leq \Delta_j \leq \frac{1}{a_j}$, and set $\Delta = \Delta(b) := \max\{\Delta_j\}$. If c satisfies $\sum_j a_j c_j \leq u$ then $\sum_j a'_j c_j \leq bu + \Delta u$: Clearly, $\sum_j a'_j c_j \leq bu + \sum_j (a'_j - ba_j) c_j$ and our claim follows from

$$0 \leq \sum_j (a'_j - ba_j) c_j = \sum_j \frac{\lceil ba_j \rceil - ba_j}{a_j} a_j c_j \leq \sum_j \Delta_j a_j c_j \leq \Delta \sum_j a_j c_j \leq \Delta u.$$

One can easily check that this bound is tight. So, we re-define the integer interval by $u' := \lfloor bu + \Delta u \rfloor$. Then, we have to decompose Δu integers in addition to the $(u - l)b$ integers we expect without rounding errors. We stress that the runtime of this approach is dominated by the number of *decompositions* of these integers, and not by the number of integers itself.

As an example, consider the alphabet CHNOPS and blowup factor $b = 10^5$, then $\Delta = \Delta_{\mathrm{H}} = 0.492936$, so for $M_0 = 1000$ we have to decompose an additional 492 integers.

3.1 Optimal blowup factor b

If we had an infinite amount of memory then we could make the blowup factor b large, thereby countering the effect of rounding error accumulation. But choosing a blowup factor b results in a table of size $O(na_1 b)$ which induces an upper bound on the blowup factor. We are left with the question how to find a good factor b that results in a small quotient $\Delta(b)/b$ of additional integers we have to decompose.

Suppose that memory considerations imply a maximal blowup factor of $B \in \mathbb{R}$. We want to find $b \in (0, B]$ such that $\Delta(b)/b$ is minimized. We can explicitly find an optimal such b by constructing the piecewise linear functions $\Delta_j(b) := \frac{1}{a_j}(\lceil ba_j \rceil - ba_j)$ with $\lceil a_j B \rceil + 1$ sampling points, for all $j = 1, \ldots, n$. Next, we set $\varphi_1 \equiv \Delta_1$ and for $j \geq 2$, we define φ_j as the maximum of φ_{j-1} and Δ_j, a piecewise linear function with $(a_1 + \cdots + a_j)B$ sampling points. Then, $\Delta \equiv \varphi_n$ is a continuous, piecewise linear function with $O((a_1 + \cdots + a_n)B)$ sampling points. We can construct Δ in time $O(n(a_1 + \cdots + a_n)B) = O(n^2 a_n B)$. For every piecewise linear part $I \subseteq \mathbb{R}$ of Δ the minima of $\Delta(b)/b$ must be located at the terminal points, so it suffices to test the $O(na_n B)$ sampling points of Δ to find the minimum of $\Delta(b)/b$.

Regarding our application of finding sum formulas over the alphabet CHNOPS, we found that choosing an optimal blowup factor has a negligible impact on runtimes. Still, the impact can be significant for other applications.

4 Scoring candidate molecules

We want to discriminate between (tens of thousands of) candidate molecules generated by decomposing the monoisotopic mass. To this end, we compare the simulated isotopic distribution with the measured peaks. Matching peak pairs between the spectra is trivial for this application.

[27] and [26] suggest to use Bayesian Statistics to evaluate mass spectra matches:

$$\mathbb{P}(\mathcal{M}_j|\mathcal{D},\mathcal{B}) = \frac{\mathbb{P}(\mathcal{M}_j|\mathcal{B})\,\mathbb{P}(\mathcal{D}|\mathcal{M}_j,\mathcal{B})}{\sum_i \mathbb{P}(\mathcal{M}_i|\mathcal{B})\,\mathbb{P}(\mathcal{D}|\mathcal{M}_i,\mathcal{B})}$$

where $\mathcal{D}$ is the data (the measured spectrum), $\mathcal{M}_i$ are the models (the candidate molecules), and $\mathcal{B}$ stands for any prior background information. In particular, we set the prior probability $\mathbb{P}(\mathcal{M}_j|\mathcal{B})$ to zero for all molecules but the decompositions of the monoisotopic mass. We can also use the abundance of certain elements to assign a low prior probability to certain molecules (say, molecules where phosphor constitutes more than 50 % of the mass). In particular, we assign prior probability zero to sum formulas that cannot correspond to a molecule, because of chemical considerations: For any molecule, the *degree of unsaturation* (DU) [17]

$$DU = -\tfrac{v_1}{2} + \tfrac{v_3}{2} + v_4 + 1 \qquad (2)$$

is a non-negative integer, where v_1, v_3, v_4 denote the number of monovalent atoms (hydrogen), trivalent atoms (nitrogen, phosphor), and tetravalent atoms (carbon) if we assume that all elements are in their lowest valency state. For higher valency states of sulfur and phosphor we may assign lower prior probabilities, as we rarely observe phosphor (sulfur) with five (six) single bonds in organic compounds.

Next, we assign probabilities to the observed masses and intensities. Assuming independence (in particular from background information) we calculate:

$$\mathbb{P}(\mathcal{D}|\mathcal{M},\mathcal{B}) = \prod_j \mathbb{P}(M_j|m_j) \prod_j \mathbb{P}(f_j|p_j) \qquad (3)$$

Here, $\mathbb{P}(M_j|m_j)$ is the probability to observe peak j at mass M_j when its true mass is m_j, and $\mathbb{P}(f_j|p_j)$ is the probability to observe peak j with intensity f_j when its true intensity is p_j. Clearly, the independence of peak intensities is violated because these intensities sum to one, but (3) can be seen as a rough estimate of the true probability.

4.1 Empirical distributions of mass and intensity differences

We want to compare the true peak masses and intensities of isotopic distributions to the experimentally determined ones. In addition to the 69 mass spectra as described in Section 5.1 we used spectra of 33 molecules with mass above 1000 Da to estimate these parameters.

Our data shows a systematic mass shift due to calibration inaccuracies, but this can be eliminated for all masses but the monoisotopic mass: We do not compare masses of the +1, ... peaks directly but instead, the difference to the monoisotopic peak, $M_j - M_0$ vs. $m_j - m_0$ for $j \geq 1$. In accordance with expert knowledge, mass differences increase with increasing mass of the molecule, so we use relative mass differences: $\Delta_0^{\mathrm{m}} := (M_0 - m_0)/m_0$ and $\Delta_j^{\mathrm{m}} := (M_j - M_0 - m_j + m_0)/m_j$ for $j = 1, 2, 3$. Confer Table 4 for mean and variance of these observations. There are only 29 +4 peaks and even fewer +5, ... peaks present in the measured mass spectra.

For intensities, our data indicates that ratios between measured and predicted peak intensity f_j/p_j follow a log normal distribution, so we determine mean and variance of $\Delta_j^{\mathrm{i}} := \log_{10} f_j - \log_{10} p_j$ for $j = 0, \ldots, 3$, confer Table 4.

	Δ_0^{m}	Δ_1^{m}	Δ_2^{m}	Δ_3^{m}
# observations	102	102	73	29
mean	$1.978 \cdot 10^{-7}$	$2.730 \cdot 10^{-7}$	$4.985 \cdot 10^{-7}$	$1.085 \cdot 10^{-6}$
std. deviation	$8.858 \cdot 10^{-7}$	$9.979 \cdot 10^{-7}$	$4.243 \cdot 10^{-6}$	$2.873 \cdot 10^{-6}$
variance	$7.847 \cdot 10^{-13}$	$9.958 \cdot 10^{-13}$	$1.800 \cdot 10^{-11}$	$8.253 \cdot 10^{-12}$

	Δ_0^{i}	Δ_1^{i}	Δ_2^{i}	Δ_3^{i}
mean	0.0111	−0.0155	−0.0809	−0.0440
std. deviation	0.02018	0.03758	0.08060	0.07682
variance	0.00041	0.00141	0.00650	0.00590

Table 4. Estimated parameters for the distribution of mass and intensity differences. See text for details.

4.2 Estimating mass and intensity probabilities

We want to estimate the probability that, given a peak with true mass m_j, we observe a peak in the measured spectrum at mass M_j: More precisely, the probability to observe a mass difference of $|M_j - m_j|$ or larger. For simplicity we assume that relative mass differences follow a Gaussian distribution with parameters $(\bar{\mu}, \bar{\sigma})$. We can then compute this probability using the complementary error function "erfc":

$$\mathbb{P}(\text{mass difference} \geq x) = \operatorname{erfc}\Big(\frac{z}{\sqrt{2}}\Big) = \frac{2}{\sqrt{2\pi}} \int_z^\infty e^{-t^2/2} dt \quad \text{with } z := \frac{|x - \bar{\mu}|}{\bar{\sigma}} \tag{4}$$

Thus, we estimate

$$\mathbb{P}(M_j | m_j) = \operatorname{erfc}\Big(\frac{|x_j - \bar{\mu}_j|}{\sqrt{2}\,\bar{\sigma}_j}\Big) \tag{5}$$

with $x_0 = (M_0 - m_0)/m_0$ and $x_j = (M_j - M_0 - m_j + m_0)/m_j$ for $j \geq 1$. Parameters $(\bar{\mu}_j, \bar{\sigma}_j)$ are listed in Table 4 where we set $\bar{\mu}_j := \bar{\mu}_3$ and $\bar{\sigma}_j := \bar{\sigma}_3$ for $j > 3$. Analogous computations can be executed for intensity differences.

Note that the distributions of mass and log intensity differences may deviate from Gaussian, and that (3) and (5) are only rough estimates. But time efficient methods are available for computing $\mathrm{erfc}(z)$ with high accuracy, so this approach may be used as a filter to find, say, 10–100 candidates that match the sample spectrum reasonably well.

4.3 Estimating missing peak probabilities

So far, we have assumed that we can detect the first K peaks of the isotopic distribution. But this is rarely the case, because peaks of small intensity are regularly lost in the "noise" of the mass spectrum. What are mass and intensity of a peak not present in the measured spectrum? We cannot estimate its mass, but we can find an upper bound for its intensity: A measured mass spectrum contains many "peak candidates", and to decide whether any such peak candidate is a "true peak", an intensity threshold is applied. So, it is reasonable to believe that, if peak $+i$ was detected in the measured spectrum with intensity f_i, then any peak $+j$ not detected in the spectrum must have intensity $f_j < f_i$ because otherwise, this peak should have been detected, too.

So, we can use the smallest intensity of the detected peaks $f_{\min}$ as an *upper* bound for the intensity of all missing peaks. We can derive tighter bounds from the measured spectrum, but we used this bound for the following evaluation. The probability to miss peak $+j$ with theoretical intensity p_j can be estimated by

$$\mathbb{P}(\text{peak } +j \text{ missing}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} dt \quad \text{with } z := \frac{x - \tilde{\mu}_j}{\tilde{\sigma}_j} \tag{6}$$

where $\tilde{\mu}_j$, $\tilde{\sigma}_j$ are the parameters of the log normal distribution of intensities and $x = \log \frac{f_{\min}}{p_j}$. Since for $j \geq 4$ we cannot derive these parameters from our data, we assume that they are identical to $\tilde{\mu}_3$, $\tilde{\sigma}_3$.

5 Computational results

5.1 Data set

Our data set consists of 69 mass spectra with single charge from several organic (macro)molecules, composed of the elements CHNOPS. For every such spectrum, the sum formula of the sample molecule is known. The spectra were acquired over the last two years; the molecules range in mass from 284 to 960 Da. Electrospray ionization (ESI) experiments were performed using a Fourier Transform Ion Cyclotron Resonance (FT-ICR) mass spectrometer APEX III (Bruker Daltonik GmbH, Bremen, Germany) equipped with a 7.0 T, 160 mm bore superconducting magnet (Bruker Analytik GmbH – Magnetics, Karlsruhe, Germany), infinity cell, and interfaced to an external (nano)ESI ion source. Peak detection and estimation of peak masses and intensities (heights) are conducted using vendor software.

5.2 Identification accuracy and runtimes

Every input "mass spectrum" consists of masses $M_0, \dots, M_k$ and intensities $f_0, \dots, f_k$. For every such spectrum, we compute all molecules such that the monoisotopic mass m_0 has relative mass difference of at most 2 ppm, $|M_0 - m_0| / m_0 \leq 2 \cdot 10^{-6}$. To do so, we decompose integer masses with some blowup $b \in \mathbb{R}$, see Sec. 3, and discard molecules with real mass outside the mass interval. Next, we discard molecules that have negative or non-integer degree of unsaturation DU, confer (2). For every such molecule, we compute its theoretical isotopic distribution (with $K = 10$) and compare it to the measured isotopic distribution as described in Section 4. We rank the molecules according to resulting probabilities. We do not use any other background information to identify the molecule, in order to be able to evaluate the discriminative power of isotopic patterns.

Out of the 69 mass spectra, 35 result in a correct identification; in 81 % of the mass spectra, the correct interpretation is found in the top 10 interpretations. There is a clear correlation between mass and identification accuracy, confer Table 5. For mass spectra below 700 Da, the correct interpretation is always found in the top 10 interpretations.

	no.	rank in output list					no. sum formulas			
mass range	spectra	1	2	3–5	6–10	11+	int.	real	chem.	runtime
200–300	3	3	0	0	0	0	60.7	26.3	5	0.0006
300–400	20	18	2	0	0	0	165.3	70.1	6.4	0.0012
400–500	25	13	5	5	2	0	560.3	236.4	17.8	0.0043
500–600	1	0	1	0	0	0	1956	833	51	0.0164
600–700	2	1	0	1	0	0	2204	934.5	30.5	0.0190
700–800	5	0	2	1	0	2	7548.6	3205.2	167.6	0.0706
800–900	8	0	1	0	1	6	12521	5325.9	340.6	0.1217
900–1000	5	0	0	0	0	5	23443	9972.8	770	0.2338

Table 5. Number of correct sum formulas at certain positions of the output list, for several mass ranges. Runtimes in seconds per spectrum. See text for details.

We analyzed all 69 mass spectra on a Pentium M 1.5 GHz processor with blowup $b = 5 \cdot 10^4$, using only a few Megabyte of memory. This results in runtimes of less than 1/4 second per spectrum for the complete analysis of one mass spectrum, including generation of molecule candidates, simulation of isotopic patterns, and ranking the measured data against the simulated pattern. Clearly, runtimes depend on molecule masses, see Table 5. Optimizing the blowup b (Sec. 3.1) did not show a significant impact on runtimes. Increasing the blowup beyond $5 \cdot 10^4$ increased runtimes: A similar behavior was observed in [5], presumably because the smaller table can be kept in the processor cache whereas the larger has to be stored and accessed in main memory.

For every mass range, we also report in Table 5 the number of integer decompositions, the number of real decompositions (cf. Sec. 3), and the number of sum formulas with non-negative integer degree of unsaturation (2). These numbers are averages over all molecules in the mass range.

6 Generalization to other elements

As noted in Section 2, we have sometimes restricted ourselves to the elements CHNOP. Regarding the natural isotopes of these elements, the isotope with smallest mass number is by far most common. For example, consider molecules consisting solely of carbon: The second isotope species exceeds the first only if 90 or more carbon atoms, with a total mass of 1080 Da or more, are present. Thus, we may assume that the monoisotopic peak can be detected and identified in the mass spectrum. Furthermore, these molecules have isotopic distributions that decrease rapidly with increasing mass.

We have ignored elements such as sulfur for the sake of brevity: These elements can have isotopic distributions that differ significantly from that of carbon, the element usually dominating a molecule's isotopic distribution. We now describe the adjustments and modifications needed for our approach to carry over to arbitrary elements. In particular, we show how to deal with sulfur-containing molecules.

The isotopic distribution of sulfur assigns lower probability to the monoisotopic molecule than even carbon: For the molecule S_{23} with monoisotopic mass 735.358 Da, the intensity of the +2 peak exceeds that of the monoisotopic peak. For the molecule S_{63} with monoisotopic mass 2014.240 Da and nominal mass $n = 2016$, the normalized intensity of the +9 peak is 4.7 % and the intensities of the $+10, +11, \ldots$ peaks sums up to 14.0%. To allow an accurate normalization of peak intensities we therefore have to take into account peaks past the +10 peak. Other elements may force us to increase K for even smaller masses. This results in increased runtimes, but no changes to our method are necessary.

Let us have another look at the isotopic distribution of the molecule S_{63}: The monoisotopic peak has a relative intensity of 4 % compared to 17% of the most intense +4 peak. So, we can detect (and decompose) the monoisotopic peak of molecules with mass up to 2000 Da that contain sulfur. In case the molecule contains other elements such as tungsten (also known as wolfram; the lightest natural isotope ^{180}W has abundance of only 0.12 %) then the monoisotopic signal will not be observable. In this case, we estimate the *average mass* of the molecule as $M_{\text{av}} := \sum_i f_i M_i$. Due to missing peaks this estimation is erroneous, but this error is superseded by measurement errors. The average mass of an element E can be estimated as the weighted sum of isotope masses, see Table 1. Then, instead of decomposing the monoisotopic mass we decompose the molecule's average mass, while the rest of our analysis remains unchanged.

If resolution and dynamic range of the mass spectrometer are very large, this may violate our assumption that the superposition of isotope species results in single +1, +2, . . . peak. For elements CHNOP we may safely ignore this fact. For sulfur, we note that the second most abundant isotope is not ^{33}S but ^{34}S with abundance of more than 4 %, and this isotope has a mass difference of 1.995796 Da that differs significantly from the mass difference of two ^{13}C isotopes, 2.00671 Da. So, mass spectrometers may detect two +2 peaks in the isotope pattern, one corresponding to a molecule having exactly one ^{34}S isotope, the other being a superposition of all remaining isotope species. To simulate this behavior, we compute the isotopic distribution of the molecule without sulfur, and the isotope species of the molecule consisting solely of sulfur. We then fold the isotope *species* and eventually merge species that cannot be differentiated due to resolution constraints. Elements that require this particular attention can be identified by experts; a rigorous formal analysis is in preparation. For the data presented in this paper, no special care was taken of sulfur because the resolution of the instrument used was not sufficient to resolve sulfur peaks.

7 Conclusion of Part I

We presented an approach to determine the sum formula of an unknown metabolite solely from its high resolution isotopic distribution. Our approach allows us to reduce the number of potential sum formulas to only a few candidates; in many cases we were able to determine the correct sum formula. The approach is time and memory efficient and can be executed on a regular desktop PC. Results on experimental data show the potential of our approach, in particular for metabolites below 700 Da.

Nevertheless, our results are only a first step towards automated determination of sum formula from high resolution mass spectrometry data. We want to conduct further studies regarding mass and intensity variations for this type of data, to achieve better discrimination between sum formula candidates. We are currently gathering an independent test set of about 100 sample spectra. Note that we have deliberately ignored some information available in the data, in order to evaluate the discriminative power of a single isotopic pattern. For example, a mass spectrum often contains different charge states of the same molecule. Also, we may use the proportion of elements in a sum formula as a prior probability for our identification: Regarding phosphor, for only 10 % of sum formulas in the KEGG LIGAND database [9] more than 18 % of the molecule's mass results from phosphor atoms. We are currently evaluating the impact of using such (background) information. We will apply our techniques to molecules that contain elements different from CHNOPS, such as selenium and silicon. [7] use ions resulting from neutral losses of the parent ion to further increase the resolving power, and we plan to extend our approach to incorporate information from neutral losses, even when multiple parent ions are present simultaneously. We also plan to process raw mass spectra, because peak picking software commonly tries to fit a peak model (Gaussian) to the data, whereas we are interested in the mean peak mass for a collection of isotope species.

Finally, we note that mass spectrometry instruments with better mass accuracy and resolution than the instrument used in our evaluation, are available these days. The development of new mass spectrometry techniques with ever increasing mass accuracy will presumably continue in the next years, and will allow us to push the mass limit for sum formula determination even further. We are currently conducting simulations to evaluate the impact of increased mass accuracy.

Part II

Decomposing metabolomic isotope patterns

8 Introduction to Part II

Mass spectrometry (MS) allows determining accurately the molecular mass of sample molecules. As with most analysis techniques in the life sciences, not one but millions of copies of the same molecule are needed. The output of a mass spectrometer, after preprocessing, consists of peaks that ideally correspond to the masses of the sample molecules and their abundance, i.e., the number of sample molecules with this mass. This brings into play the natural isotopic distributions of the elements: Several peaks in the output correspond to the same type of sample molecule, reflecting its isotope pattern. In this paper, we make use of this isotope pattern to identify the sample molecule.

Metabolites, such as sugars or lipids, are small molecules that are intermediates or products of the metabolism and that participate in most processes of the cell. Yet, to date most remain uncharacterized. Large metabolite libraries exist but their use is limited to identifying metabolites that are already known. High resolution mass spectrometry allows to determine the mass of a sample molecule with very high accuracy (up to 10^{-3} Dalton), and has become one preferred method of analyzing metabolites. When trying to identify a metabolite, the first and most crucial step is determining its sum formula, i.e., the number of atoms of each element.

Our input is a list of masses $M_0, \ldots, M_K$ with intensities[6] $f_0, \ldots, f_K$, normalized such that $\sum_i f_i = 1$. We assume that these have been extracted from a mass spectrum in a preprocessing step, and that they correspond to the isotope pattern of a sample molecule.[7] Our goal then is to find the molecule, or rather its sum formula, whose isotope pattern best matches the input. In the following, we use "molecule" and "sum formula" interchangeably.

One way to solve this problem is by computing all molecules with monoisotopic mass sufficiently close to M_0, simulating their isotope pattern, and matching it with the input. However, the number of molecules with a certain mass increases rapidly for large masses, see Section 11. Thus, it is essential to find methods for fast simulation of isotope patterns. This problem has previously been addressed e.g. in [8,25]. Here we present a method for rapid computation of isotope distributions and, in particular, mean masses of isotope peaks, improving on results in [20].

Even more importantly, methods for reducing the search space are needed. The problem of determining the sum formula of a sample molecule was addressed frequently from the biochemical and mass spectrometry viewpoint [3, 6, 18, 23]. It can be stated in mathematical terms as follows: Given σ positive numbers $a_1, \ldots, a_\sigma$ and a query M, find a non-negative integer vector $(c_1, \ldots, c_\sigma)$ such

[6] The height of the peaks is referred to as "intensity" (of the signal). Note that high resolution mass spectrometry allows for such high accuracy within a small range that, as opposed to most other MS applications, here the intensities of the peaks can be relied upon, and the isotopic peaks can be well separated.

[7] Note that, for molecular mixtures, separating isotopic peaks that belong to different molecules is trivial in this case.

that $\sum_i c_i a_i = M$. Here, $a_1, \ldots, a_\sigma$ correspond to the masses of the elements and M to the mass of the sample molecule. This is an Integer Knapsack Problem; the variant where the a_i are positive integers is also known as Coin Change Problem. Both are NP complete, and can be solved by a simple dynamic programming algorithm in pseudo-polynomial time.

We employ an algorithm introduced in [4] for computing *all* solutions c, which is greatly superior to simple backtracking in the classic dynamic programming table both in its time and space requirements. We develop certain pruning conditions which we employ during runtime, and which successfully reduce the search space, discarding many candidates before they are computed. To this end, we introduce the problem of jointly decomposing a set of queries. These are *not* the input masses $M_0, \ldots, M_k$, but other values derived from the input such as intensities or average mass, for which we define appropriate weighted alphabets. Details of how to postprocess and rank the remaining candidates can be found in Part I.

The problem of deriving sum formulas from isotope patterns has recently been investigated in [7,11,16], but these studies concentrate on the experimental side of the problem. The authors of [11] disregard mean peak masses; computational methods are only given in [7], however, the descriptions do not yield themselves to runtime comparisons. For runtime comparisons of the decomposition algorithm and the classical DP algorithm on the amino acid, nucleotide, and CHNOPS alphabets (the latter used in this paper), see [13, Sec. 4.6]. In this paper, we give experimental results using data extracted from the KEGG LIGAND database [9].

The paper is organized as follows. We give the necessary physical background in Section 9. We introduce our model in Section 10 and show how to generate isotope patterns efficiently. After a brief sketch of the decomposition problem (Section 11), we show how to extract a joint decomposition problem from the input (Section 12) and discuss joint decompositions and how to solve them in Section 13. Finally, in Section 14, we provide first experimental results.

9 Isotope species

Atoms are composed of electrons, protons, and neutrons. The number of protons (the atomic number) defines what element the atom is. The elements most abundant in living beings are hydrogen (symbol H) with atomic number 1, carbon (C, 6), nitrogen (N, 7), oxygen (O, 8), phosphor (P, 15), and sulfur (S, 16). The number of neutrons, on the other hand, can vary: Atoms with the same number of protons but different numbers of neutrons are called *isotopes* of the element. For example, hydrogen has two natural isotopes (i.e., isotopes that occur in nature), ^{1}H and ^{2}H (deuterium): ^{1}H consists of one proton and one electron, while ^{2}H consists of one proton, one electron, and one neutron. Each of these isotopes occurs in nature with a certain abundance. The superscript preceding the symbol denotes the *mass number* of the atom: the number of protons plus the number of neutrons. Regarding the other elements listed above, carbon and nitrogen have two natural isotopes, oxygen has three, sulfur four, and phosphor occurs in only one isotopic type.

The *mass* of an atom is measured in Dalton (Da), which is defined as one twelfth of the mass of a ^{12}C isotope.[8] An atom's mass is roughly but not exactly equal to its mass number, the difference being due to the binding energy in the atom's nucleus. The masses of the different isotopes and their abundance are known up to very high precision; for example, ^{1}H has mass 1.007825 Da with abundance 99.985%, and ^{2}H mass 2.014102 Da with abundance 0.015%. See Part I, Section 2 for an isotope table of the six elements listed above, and [1] for a complete table.

The *nominal mass* (also called *nucleon number*) of a molecule is the sum of protons and neutrons of the constituting atoms. The *mass* of the molecule is the sum of masses of these atoms. Clearly, nominal mass and mass depend on the isotopes the molecule consists of, thus on the *isotope species (isobars)* of the molecule. The isotope species where each atom is the isotope with the lowest nominal mass is called *monoisotopic.* Likewise, the mass of the monoisotopic species is called the *monoisotopic mass* of the molecule. For example, sucrose $C_{12}H_{22}O_{11}$ has monoisotopic mass 342.116215 Da with monoisotopic nominal mass 342. We note that metabolites are "rather small" molecules with mass seldom exceeding 1000 Da.[9]

The number of isotope species with distinct mass for a molecule with i_{H} hydrogen, i_{C} carbon, i_{N} nitrogen, i_{O} oxygen, i_{P} phosphor, and i_{S} sulfur atoms is

$$\text{number of isotope species} = (i_{\mathrm{C}}+1)(i_{\mathrm{H}}+1)(i_{\mathrm{N}}+1)\tbinom{i_{\mathrm{O}}+2}{2}\tbinom{i_{\mathrm{S}}+3}{3}, \tag{7}$$

if we assume that all mass differences are linearly independent over the rational numbers. This follows because for an element E with r isotope types, a molecule E_l consisting of l atoms of the element has $\binom{l+r-1}{r-1}$ different isotope species.

The probability that a certain isotope species occurs can be computed by multiplying the probabilities of the underlying isotopes. See Table 6 for the first eleven isotope species of sucrose. In total, sucrose has $13 \cdot 23 \cdot \binom{13}{2} = 23\,322$ isotope species.

Given the isotope species of two molecules, we can easily calculate the isotope species of the joined molecule by folding the species: Species with masses m_1, m_2 and probabilities p_1, p_2 add a contribution of $p_1 p_2$ to the isotope species with mass $m_1 + m_2$ in the joined molecule.

We will refer to the set of elements as our *alphabet Σ*, and to the six elements mentioned above, simply as CHNOPS.

[8] Dalton is the unit commonly used in molecular biology and biochemistry, while in physics, the same quantity is denoted "u" (unified atomic mass unit).

[9] In the KEGG LIGAND database, 95, 6% of sum formulas have mass below 1000 Da.

^{12}C	^{13}C	^{1}H	^{2}H	^{16}O	^{17}O	^{18}O	nom. mass	mass (Da)	abundance %
12	0	22	0	11	0	0	342	342.116215	84.9204
11	1	22	0	11	0	0	343	343.119570	11.4384
12	0	22	0	10	1	0	343	343.120431	0.3558
12	0	21	1	11	0	0	343	343.122492	0.2803
12	0	22	0	10	0	1	344	344.120460	1.8727
10	2	22	0	11	0	0	344	344.122925	0.7062
11	1	22	0	10	1	0	344	344.123786	0.0479
11	1	21	1	11	0	0	344	344.124647	0.0007
12	0	22	0	9	2	0	344	344.125847	0.0378
12	0	21	1	10	1	0	344	344.126708	0.0012
12	0	20	2	11	0	0	344	344.128769	0.0004

Table 6. Isotope species of sucrose molecules $C_{12}H_{22}O_{11}$, sorted by mass. Isotope species with nominal mass ≥ 345 omitted.

10 Isotope patterns

No present-day analysis technique is capable of resolving isotope species with identical nominal mass. Instead, these isotope species appear as one single peak in the MS output.[10] For this reason, we merge isotope species with identical nominal mass; we refer to the resulting distribution as the molecule's *isotope pattern*.

For each element $E \in \Sigma$ we define two discrete random variables, denoted X_E and Y_E, representing the mass and the mass number, respectively. For example, X_{C} with state space $\{12, 13.003355\}$ and Y_{C} with state space $\{12, 13\}$ and

$$\mathbb{P}\big(X_{\mathrm{C}} = 12\big) = \mathbb{P}\big(Y_{\mathrm{C}} = 12\big) = 0.98890,$$
$$\mathbb{P}\big(X_{\mathrm{C}} = 13.003355\big) = \mathbb{P}\big(Y_{\mathrm{C}} = 13\big) = 0.01110$$

are the random variables of carbon. Given a molecule consisting of l atoms, we assign to the ith atom, $i = 1, \ldots, l$, two random variables X_i and Y_i, where $X_i \sim X_E$ and $Y_i \sim Y_E$, with E being the corresponding element. Now we can represent the molecule's *mass distribution* by the random variable $X := X_1 + \ldots + X_l$, and its nominal mass distribution, or *isotopic distribution*, by $Y := Y_1 + \ldots + Y_l$. Note that X and Y are correlated, since X_E can be viewed as a function of Y_E and E.

In an ideal mass spectrum, normalized peak intensities correspond to the isotopic distribution of the molecule. For ease of exposition, the peak at monoisotopic mass is also called monoisotopic, the following peaks are referred to as +1, +2, ... peaks. See Table 7 for the isotopic distribution of sucrose.

It is important to observe that regarding the six elements most abundant in living beings, all resulting molecules have isotopic distributions that decrease

[10] The case of sulfur-containing molecules is an exception and needs special attention, we omit the details.

rapidly with increasing mass. In particular, we can restrict ourselves to computing the first K non-zero values of the distribution, for rather small K such as $K = 10$. For example, consider the molecule C_{166} with nominal mass 1992: The intensities of $+10, +11, \ldots$ peaks sum up to less than 0.00003.

10.1 Computing the isotopic distributions of E_l

The atoms hydrogen, carbon, and nitrogen have only two isotopes. Thus, the isotopic distribution of a molecule E_l consisting of l identical atoms of type E with $E \in \{\mathrm{H}, \mathrm{C}, \mathrm{N}\}$ follows a binomial distribution: Let q_k denote the probability that E_l has nominal mass $n+k$, where n is the monoisotopic nominal mass of E_l. Then, $q_k = \binom{l}{k} p^{l-k}(1-p)^k$ where p is the probability of the monoisotopic isotope. The values of the q_k can be computed iteratively, since $q_{k+1} = \frac{l-k}{k+1} \cdot \frac{1-p}{p} q_k$ for $k \geq 0$, thus computation time is $O(K + \log l)$ if we compute $q_0 = p^l$ using $\log l$ multiplications.

Where an element E has $r > 2$ isotopes (such as oxygen and sulfur), the isotopic distribution of E_l can be computed as follows: Let p_i for $i = 0, \ldots, r$ denote the probability of occurrence of the ith isotope.

$$\mathbb{P}(E_l \text{ has nominal mass } n + k\ ) = \sum \tbinom{l}{l_0, l_1, \ldots, l_r} \cdot \prod_{i=0}^{r} p_i^{l_i}, \tag{8}$$

where the sum runs over all $l_0, \ldots, l_r \geq 0$ satisfying $\sum_{i=0}^r l_i = l$ and $\sum_{i=1}^r i \cdot l_i = k$ [8].

How do we find all tuples $(l_0, \ldots, l_r)$ that satisfy both conditions $\sum l_i = l$ and $\sum i \cdot l_i = k$? Those satisfying $\sum_i i \cdot l_i$ are the integer partitions of k into at most r parts, which can be computed recursively with a greedy approach. However, this approach faces the problem that the number of partitions grows rapidly, at least with a polynomial in k of degree $r - 1$ [22].

10.2 Folding isotopic distributions

Given two discrete random variables Y and Y' with state spaces $\Omega, \Omega' \subseteq \mathbb{N}$, we can compute the distribution of the random variable $Z := Y + Y'$ by folding the distributions, $\mathbb{P}(Z = n) = \sum_k \mathbb{P}(Y = k) \cdot \mathbb{P}(Y' = n - k)$. If we restrict ourselves to the first K values of this sum, we can compute this distribution in time $O(K^2)$. Kubinyi [12] suggests to compute the isotopic distributions of oxygen O_l and sulfur S_l by successive folding of the respective distribution: Using a Russian multiplication scheme for the folding, this results in an algorithm with runtime $O(K^2 \log l)$. For molecules consisting of different elements, we first compute the isotopic distributions of the individual elements, and then combine these distributions by folding in $O(|\Sigma| \cdot K^2)$ time.

Finally, note that we can use Fourier transforms of atom distributions, and instead of folding these distributions multiply the Fourier transforms [19]. Doing so we can eventually replace the K^2 factor in the algorithm's runtime by a

$K \log K$ factor. As we limit our attention to small K such as $K = 10$, this will not result in a speedup of the algorithm. In practice, this approach may face the problem of numerical errors.

10.3 Isotope peak masses

As we have seen, the imperfection of mass spectrometry results in $+1, +2, \ldots$ isotope peaks that, in fact, are superpositions of peaks with almost identical mass. What is the mass of such a superposition peak? It is reasonable to assume that its mass is the mean mass of all isotope species that add to its intensity [20]. Formally, we define a mass function $\tilde{\mu} : \mathbb{N} \to \mathbb{R}$ that maps the mass numbers of the different isotopes[11] to the corresponding real masses: $\tilde{\mu}(1) = 1.007825$, $\tilde{\mu}(2) = 2.014102, \ldots, \tilde{\mu}(34) = 33.967867$, $\tilde{\mu}(36) = 35.967081$.[12] Thus, $X_E = \tilde{\mu}(Y_E)$ for all elements E. Let the mass distribution $X = X_1 + \ldots + X_l$ and isotopic distribution $Y = Y_1 + \cdots + Y_l$ of a molecule with monoisotopic nominal mass n be given. Then, the mean peak mass of the $+k$ peak is:

$$m_k = \mathbb{E}(X \mid Y = n+k) = \sum_{\sum_i n_i = n+k} \frac{\mathbb{P}(Y_1 = n_1, \ldots, Y_l = n_l)}{\mathbb{P}(Y = n+k)} \big(\tilde{\mu}(n_1) + \cdots + \tilde{\mu}(n_l)\big). \tag{9}$$

See Table 7 for mean peak masses of sucrose. We refer to the isotopic distribution together with the mean peak masses as the molecule's *isotope pattern.*

nominal mass	342	343 (+1)	344 (+2)	345 (+3)	346 (+4)
abundance %	84.9204	12.0745	2.66683	0.297583	0.0370679
mean peak m.	342.116215	343.119663	344.121254	345.124197	346.126084

Table 7. Isotope pattern (isotopic distribution and mean peak masses) of sucrose $C_{12}H_{22}O_{11}$. Peaks with nominal mass 347 and above have abundance $< 0.01\%$.

Computing the mean peak mass using (9) is highly inefficient, because we have to sum up over all isotope species, so pruning strategies have been developed that lead to a loss of accuracy [20, 25]. But there exists a simple recurrence for computing these masses analogous to the folding of distributions, generalizing and improving on results in [20]:

Let $Y = Y_1 + \cdots + Y_l$ and $Y' = Y'_1 + \cdots + Y'_L$ be isotopic distributions of two molecules with monoisotopic nominal masses n and n', respectively. Let $p_k := \mathbb{P}(Y = n + k)$ and $q_k := \mathbb{P}(Y' = n' + k)$ denote the corresponding probabilities, m_k and m'_k the mean peak masses of the $+k$ peaks. Consider the random variable $Z = Y + Y'$ with monoisotopic nominal mass $\tilde{n} = n + n'$.

[11] See Table 1 in Part I I, Section 2.

[12] This is only possible because there exist no overlaps in mass numbers between distinct elements. For other sets of elements such overlaps do exist and we need a formally more complicated setup to define our mean peak masses. Still and all, the results of this section remain valid.

Theorem 1. *The mean peak mass $\tilde{m}_k$ of the $+k$ peak of the random variable $Z = Y + Y'$ can be computed as:*

$$\tilde{m}_k = \frac{1}{\sum_{j=0}^{k} p_j q_{k-j}} \cdot \sum_{j=0}^{k} p_j q_{k-j} \left(m_j + m'_{k-j} \right) \tag{10}$$

Proof. Note that $\sum_{j=0}^{k} p_j q_{k-j} = \mathbb{P}(Z = \tilde{n} + k)$. Let $\boldsymbol{n} = (n_1, \ldots, n_l) \in \mathbb{N}^l$ and $\boldsymbol{n'} = (n'_1, \ldots, n'_L) \in \mathbb{N}^L$ be vectors of nominal masses. We denote $\sum \boldsymbol{n} := \sum_{i=1}^{l} n_i$ and $\sum \boldsymbol{n'} := \sum_{i=1}^{L} n'_i$. Let $\boldsymbol{Y} := (Y_1, \ldots, Y_l)$ and $\boldsymbol{Y'} := (Y'_1, \ldots, Y'_L)$ be vectors of the input random variables, and note that

$$\mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}, \boldsymbol{Y'} = \boldsymbol{n'}) = \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}) \mathbb{P}(\boldsymbol{Y'} = \boldsymbol{n'})$$

due to the independence of the underlying random variables. Finally, we set $\tilde{\mu}(\boldsymbol{n}) = \sum_{i=1}^{l} \tilde{\mu}(n_i)$ and analogously define $\tilde{\mu}(\boldsymbol{n'})$. We can rewrite (9) for the mass of the $+k$ peak as

$$\mathbb{P}(Z = \tilde{n} + k) \cdot \tilde{m}_k = \sum_{\sum \boldsymbol{n} + \sum \boldsymbol{n'} = \tilde{n} + k} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}, \boldsymbol{Y'} = \boldsymbol{n'}) \cdot \big(\tilde{\mu}(\boldsymbol{n}) + \tilde{\mu}(\boldsymbol{n'}) \big).$$

We observe that we can split this formula into two independent sums of the form

$$\sum_{\sum \boldsymbol{n} + \sum \boldsymbol{n'} = \tilde{n} + k} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}, \boldsymbol{Y'} = \boldsymbol{n'}) \cdot \tilde{\mu}(\boldsymbol{n}) \tag{11}$$

and a second summand where $\tilde{\mu}(\boldsymbol{n})$ is replaced by $\tilde{\mu}(\boldsymbol{n'})$; we concentrate on (11) in the following. Now,

$$\begin{aligned}
&\sum_{\sum \boldsymbol{n} + \sum \boldsymbol{n'} = \tilde{n} + k} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}, \boldsymbol{Y'} = \boldsymbol{n'}) \cdot \tilde{\mu}(\boldsymbol{n}) \\
&\qquad = \sum_{j=0}^{k} \sum_{\sum \boldsymbol{n} = n + j} \sum_{\sum \boldsymbol{n'} = n' + k - j} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}) \mathbb{P}(\boldsymbol{Y'} = \boldsymbol{n'}) \cdot \tilde{\mu}(\boldsymbol{n}) \\
&\qquad = \sum_{j=0}^{k} \sum_{\sum \boldsymbol{n} = n + j} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}) \cdot \tilde{\mu}(\boldsymbol{n}) \sum_{\sum \boldsymbol{n'} = n' + k - j} \mathbb{P}(\boldsymbol{Y'} = \boldsymbol{n'}) \\
&\qquad = \sum_{j=0}^{k} \sum_{\sum \boldsymbol{n} = n + j} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}) \cdot \tilde{\mu}(\boldsymbol{n}) \cdot \mathbb{P}(Y'_1 + \cdots + Y'_L = n' + k - j) \\
&\qquad = \sum_{j=0}^{k} \mathbb{P}(Y' = n' + k - j) \sum_{\sum \boldsymbol{n} = n + j} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{n}) \cdot \tilde{\mu}(\boldsymbol{n}) \\
&\qquad = \sum_{j=0}^{k} q_{k-j} p_j m_j
\end{aligned}$$

where the last equality follows from the definition of m_j,

$$m_j = \frac{1}{p_j} \sum_{\sum \boldsymbol{n}=n+j} \mathbb{P}(\boldsymbol{Y}=\boldsymbol{n}) \cdot \tilde{\mu}(\boldsymbol{n}).$$

Analogously, we can show that

$$\sum_{\sum \boldsymbol{n}+\sum \boldsymbol{n'}=\tilde{n}+k} \mathbb{P}(\boldsymbol{Y}=\boldsymbol{n}, \boldsymbol{Y'}=\boldsymbol{n'}) \cdot \tilde{\mu}(\boldsymbol{n'}) = \sum_{j=0}^{k} q_{k-j} p_j m'_j.$$

This concludes the proof of the theorem.

The theorem allows us to "fold" mean peak masses of two distributions to compute the mean peak masses of their sum. This implies that we can compute mean peak masses as efficiently as the distribution itself, confer the previous section.

11 Integer decompositions

Determining the sum formula of a molecule from its mass M amounts to writing M as a non-negative integer linear combination of the masses of the individual atoms, or finding a *decomposition* of M over these masses.

Let for a moment both the masses $\{a_1, \ldots, a_\sigma\}$ of the alphabet Σ and the query mass m be positive integers. We are looking for a non-negative integer vector $(c_1, \ldots, c_n)$ such that $\sum_i c_i a_i = m$. This is a well-studied problem, referred to in its different variants as Coin Change Problem, Change Making Problem, or Money Changing Problem, and can be solved with a simple dynamic programming algorithm in pseudo-polynomial time [14]. Recently, two of the authors presented a novel algorithm for determining all such decompositions [4], with runtime $O(a_1 \sigma \gamma(m))$ and space $O(a_1 \sigma)$, where σ is the size of the alphabet, a_1 is the smallest mass and $\gamma(m)$ the number of decompositions of m.

We briefly sketch the algorithm. Given an integer alphabet $a_1 \leq \ldots \leq a_\sigma$ relatively prime, a data structure of size σa_1, referred to as *Extended Residue Table* (ER table), is computed in a preprocessing step. Entry $\mathrm{ER}(r, i)$, for $r = 0, \ldots, a_1 - 1$ and $i = 1, \ldots, \sigma$, is the smallest number congruent r modulo a_1 which is decomposable over the alphabet $\{a_1, \ldots, a_i\}$. Thus, the last column $\mathrm{ER}(\cdot, \sigma)$ of the table gives, for each residue r, the smallest number congruent r modulo a_1 that is decomposable over the given alphabet. Computation time is $O(\sigma a_1)$, using a modification of the Round Robin Algorithm introduced in [5]. All decompositions of the query m are then recursively generated, limiting the number of unsuccessful paths by using information from the ER table. As a result, the runtime of the algorithm is proportional only to the size of the table σa_1 and the number of decompositions $\gamma(m)$, and does not depend directly on the input m itself.

For decomposing molecule masses, this decomposition technique has several advantages over classical dynamic programming, such as improved runtimes and favorable preprocessing. However, the main advantage of this method is the strongly reduced memory requirement that drops by a factor of about one thousand.

In order to be able to employ the algorithm, the masses need to be scaled to integers, using some precision δ. Moreover, when interpreting a mass M from the input, measurement errors have to be accounted for, thus we have to search for decompositions in the interval $[M, M+\varepsilon]$ for some ε depending on the mass spectrometer. To avoid rounding error accumulation, δ is usually set one to two orders of magnitude smaller than ε. Further non-trivial rounding error problems need to be carefully considered and eliminated; details are discussed in Part I, Section 3.

The number of decompositions $\gamma(m)$ for an integer mass m over $\{a_1, \dots, a_\sigma\}$ grows rapidly with increasing m, and asymptotically behaves like a polynomial of degree $\sigma - 1$ (Schur's Theorem [24]):

$$\gamma(m) \sim \frac{1}{(\sigma-1)!\, a_1 \cdots a_\sigma} m^{\sigma-1}. \tag{12}$$

For our alphabet CHNOPS this implies that the number of molecules with real mass in the interval $[M, M+\varepsilon]$ asymptotically behaves like $3.10657 \cdot 10^{-9} \cdot \varepsilon\, M^5$. Note that this is a rather crude approximation of the true number of decompositions as convergence is slow. A closer approximation is given in [2], of which the first few terms are:

$$\frac{1}{a_1 \cdots a_\sigma} \left(\frac{m^{\sigma-1}}{(\sigma-1)!} + \frac{m^{\sigma-2}}{2(\sigma-2)!} \sum_{i=1}^{\sigma} a_i + \frac{m^{\sigma-3}}{4(\sigma-3)!} (\frac{1}{3} \sum_{i=1}^{\sigma} a_i^2 + \sum_{i<j} a_i a_j) \right). \tag{13}$$

In Figure 2, we plot the number of decompositions for masses of up to 2000 Da over the alphabet CHNOPS, and show that Equation (13) gives a very good approximation of $\gamma(m)$:

We have computed the number of decompositions of a monoisotopic mass over the weighted alphabet CHNOPS using a classic dynamic programming approach described in [4]. Masses were rounded to integers with a precision of $\delta = 10^{-6}$ Da. Then, for every interval of width 0.001 Da we computed the number of decompositions in this interval. The resulting numbers vary due to the combinatorial structure of the problem, see the inlay in Fig. 2. For visualization, we then computed the minimal and maximal number of decompositions for every interval of width 1 Da, the resulting functions can be found in Fig. 2. We also plot the two approximations from Section 11, Equations (12) and (13). Unlike the alphabet of amino acids [4, Fig. 8] the number of decompositions over the CHNOPS alphabet does not vary strongly, that is, similar masses also have similar numbers of decompositions. This is due to the presence of hydrogen with mass one order of magnitude smaller than all other masses.

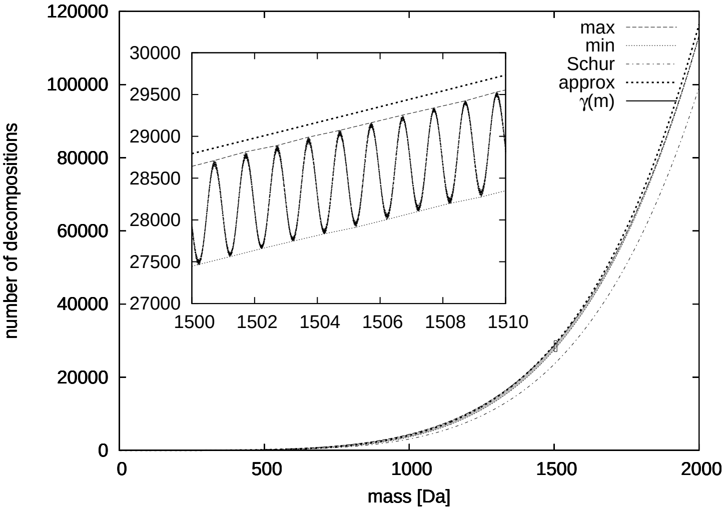

Fig. 2. Number of decompositions over the weighted alphabet CHNOPS. Interval width 0.001 Da, minima and maxima taken in intervals of width 1 Da. The true number $\gamma(m)$ of decompositions in comparison with the asymptotic formula given in Eq. (12) (Schur) and the approximation of Eq. (13) (approx). As is shown in the inlay, $\gamma(m)$ varies with a periodic function of period approx. 1 Da.

12 Additive invariants

The mass of the monoisotopic peak is an *additive invariant* of the decompositions we are searching for: Given any solution, the sum of monoisotopic masses of all elements is the input mass M_0. In this section, we present other additive invariants for molecules resulting from the observed isotopic distribution. For the following, we define a *weighted alphabet* (Σ, μ) as an alphabet Σ together with a mass function $\mu : \Sigma \to \mathbb{N}$. For simplicity, we often write $\{\mu(s_i) \mid s \in \Sigma\}$ for (Σ, μ). For the alphabet CHNOPS, we have already defined one mass function: $\mu(E)$ denotes the monoisotopic mass of element E. We will now define other mass functions for the same alphabet.

In the rest of this section, we consider a theoretical molecule where i_E denotes the multiplicity of element E in the molecule, $E \in \Sigma$. Recall that we can decompose integers only, so we assume in the following that all masses are rounded using appropriate precisions. We also ignore measurement inaccuracies and refer the reader to Part I for a detailed discussion of how to deal with these problems.

12.1 Average mass of the molecule

Given the observed normalized intensities $f_0, \ldots, f_K$ and peak masses $M_0, \ldots, M_K$, we easily estimate the average mass of the molecule as $M_{\text{av}} :=$

$\sum_i f_i M_i$. This will underestimate the average mass of the molecule, but this error is superseded by measurement errors. The average mass of an element E can be estimated by $\mathbb{E}(X_E)$. Let μ_1 denote the corresponding weight function; we decompose the number M_{av} over these weights.

12.2 Intensity of the monoisotopic peak

For every element E, let p_E denote the probability that an isotope of this element is monoisotopic. What is the intensity of the monoisotopic peak of our molecule? Clearly, this is the probability that the molecule has monoisotopic mass, which implies that all atoms must have monoisotopic mass:

$$p^* := \mathbb{P}(\text{molecule has monoisotopic mass}) = \prod_{E \in \Sigma} p_E^{i_E} \tag{14}$$

Recall that $f_0 \in [0,1]$ denotes the observed normalized intensity of the monoisotopic peak, so the measurement f_0 should agree with p^*; taking the logarithm we find

$$\sum_{E \in \Sigma} i_E \cdot \log p_E = \log f_0. \tag{15}$$

Defining a third set of weights for our alphabet, $\mu_2(E) := -\log p_E$ for every element E, we can decompose the number $-\log f_0$ over these weights. Note that by definition, $\mu_2(\mathrm{P}) = 0$ holds for phosphor.

12.3 Relative intensity of the +1 peak

Let q_E denote the probability that an isotope of this element has nominal mass one above the monoisotopic, for every element E. Note that $q_E = 1 - p_E$ for $E \in \{\mathrm{C}, \mathrm{H}, \mathrm{N}\}$, $q_E < 1 - p_E$ for $E \in \{\mathrm{O}, \mathrm{S}\}$, and $q_{\mathrm{P}} = 0$.

What is the probability that exactly one carbon atom is of isotopic type +1, while all other atoms of our molecule are monoisotopic? One can easily see that this probability is $i_{\mathrm{C}} \frac{q_{\mathrm{C}}}{p_{\mathrm{C}}} p^*$, see (14) for the definition of p^*. In total, the probability to find exactly one atom of the molecule of isotopic type +1 and, hence, the intensity of the +1 peak, is

$$\mathbb{P}(\text{molecule has nominal mass } n+1) = \textstyle\sum_{E \in \Sigma} i_E \frac{q_E}{p_E} p^*. \tag{16}$$

Recall that $f_1 \in [0,1]$ denotes the normalized intensity of the +1 peak, then comparison to the monoisotopic peak leads to the equality:

$$\sum_{E \in \Sigma} i_E \cdot \frac{q_E}{p_E} = \frac{f_1}{f_0} \tag{17}$$

Hence, we can define a fourth set of weights for our alphabet, $\mu_3(E) := q_E / p_E$ for every element E. We can decompose the number f_1/f_0 over these weights. Note that again $\mu_3(\mathrm{P}) = 0$ holds.

12.4 Mass of the +1 peak

Let $Y := Y_1 + \cdots + Y_l$ be the random variable corresponding to our molecule with monoisotopic nominal mass n. We calculate the difference between expected masses of +1 peak and monoisotopic peak, see (9) for the expected mass m_1 of the +1 peak. Let δ_E be the mass difference between the +1 mass and monoisotopic mass of element E, for example $\delta_{\mathrm{C}} = 13.003355 - 12 = 1.003355$. For phosphor we define $\delta_{\mathrm{P}} := 0$. Then,

$$m_1 - m_0 = \frac{\mathbb{P}(Y = n)}{\mathbb{P}(Y = n+1)} \sum_{E \in \Sigma} i_E \frac{q_E}{p_E} \delta_E, \tag{18}$$

where $\mathbb{P}(Y = n) = p^*$. Recall that M_0, M_1 denote the observed masses of the monoisotopic and +1 peak. The measured mass difference $M_1 - M_0$ should agree with $m_1 - m_0$, and in view of $\mathbb{P}(Y = n+1) = f_1$ and $\mathbb{P}(Y = n) = f_0$, we infer

$$\frac{f_1}{f_0} \cdot (M_1 - M_0) = \sum_{E \in \Sigma} i_E \cdot \frac{q_E}{p_E} \delta_E. \tag{19}$$

Hence, we can define a fifth set of weights for our alphabet, $\mu_4(E) := \frac{q_E}{p_E}\delta_E$ for every element E. We can decompose the number $\frac{f_1}{f_0}(M_1 - M_0)$ over these weights. Again, $\mu_4(\mathrm{P}) = 0$ holds.

13 Joint decompositions

For the current problem, we need to find *joint* decompositions for two or more masses $m_1, \ldots, m_k$ where each mass is decomposed over a different weighted alphabet of the same size. Formally, we state the

> Joint Decomposition Problem.
> Let $\{a_{1,1}, \ldots, a_{1,\sigma}\}, \ldots, \{a_{k,1}, \ldots, a_{k,\sigma}\}$ be k weighted alphabets of non-negative integers. Let $m_1, \ldots, m_k \in \mathbb{N}$. Find all joint decompositions c of $m_1, \ldots, m_k$, i.e., all $c = (c_1, \ldots, c_\sigma) \in \mathbb{N}^\sigma$ such that $Ac = m$, where $A = (a_{ij})_{i=1,\ldots,k, j=1,\ldots,\sigma}$ and $m = (m_1, \ldots, m_k)$.

The problem is also known as multidimensional integer knapsack problem. In general, it is NP complete to decide if there exists at least one solution when the matrix has integer entries [21]. At the other extreme, if we have σ many equations, then A is a square matrix, and if its rows are linearly independent, we can compute its inverse A^{-1}. We then only need to check whether $c = A^{-1}m$ has only non-negative integer entries; if this is the case, then c is a joint decomposition of $m_1, \ldots, m_\sigma$.

Using decomposition techniques of Section 11, a naïve approach to solve the joint decomposition problem is to generate all decompositions c of m_1 and then test whether $\sum_i c_i a_{j,i} = m_j$ for all $j = 2, \ldots, k$. However, this involves generating many decompositions unnecessarily. Another approach is to construct ER tables for all alphabets. Then, while running the algorithm on the ER table for alphabet $\{a_{1,1}, \ldots, a_{1,\sigma}\}$, in each step of the recursion, we check whether there is still a feasible solution for all m_j, $j = 2, \ldots, k$, as well. If the answer is negative for one j, we terminate the current recursion step and continue with the next candidate. Note that this is a runtime heuristic only, since there may exist decompositions over each alphabet, but they may contradict each other.

Consider matrix A of dimension $(k \times \sigma)$. By Gaussian elimination, we can find a lower triangular matrix $L \in \mathbb{R}^{k \times k}$ of full rank, and an upper triangular matrix $R \in \mathbb{N}^{k \times \sigma}$ such that $A = LR$. Then, $Ac = m$ if and only if $Rc = m'$, where we can compute $m' = L^{-1}m$. In particular, c must satisfy the bottom equation of $Rc = m'$:

$$0 \cdot c_1 + \cdots + 0 \cdot c_{k-1} + r_{k,k}c_k + \cdots + r_{k,\sigma}x_\sigma = m'_\sigma, \tag{20}$$

which has at most $\sigma - k + 1$ non-zero coefficients. If all coeffiecents of R are non-negative integers, we have a new instance of the joint decomposition problem, which we can solve iteratively, beginning with the bottom equation: We build ER tables for each (new) weighted alphabet, run the decomposition algorithm on the bottom one, checking in each step of the recursion whether the solution is feasible over all alphabets. When having computed a decomposition of m'_σ over alphabet $\{r_{k,k}, \ldots, r_{k,\sigma}\}$, we continue with the next equation, which has one variable more. In view of (12), the number of solutions of Equation (20) is considerably lower than of any of the original equations, so we improve on runtime.

However, even though we can guarantee that all entries of R are integers, some could be negative, yielding infinitely many solutions. In order to avoid negative entries, one needs to exchange columns, details will be described elsewhere. We describe the algorithm for two equations below.

We refer to this algorithm as Dimension Reduction (DR) algorithm. In Section 14, we will see that the DR algorithm yields a significant improvement over the approach of simulateously decomposing over the individual alphabets.

13.1 DR algorithm for two alphabets

Consider a joint decomposition problem over two weighted alphabets:

$$\begin{aligned} a_{1,1}c_1 + a_{1,2}c_2 + \ldots + a_{1,\sigma}c_\sigma &= m_1 \\ a_{2,1}c_1 + a_{2,2}c_2 + \ldots + a_{2,\sigma}c_\sigma &= m_2 \end{aligned} \tag{21}$$

Find column j such that $\frac{a_{1,j}}{a_{2,j}}$ is minimal, and exchange columns 1 and j, renaming coefficients. Thus, we have $\frac{a_{1,1}}{a_{2,1}} \leq \frac{a_{1,i}}{a_{2,i}}$ for all i. Now, applying Gaussian elimination, we can transform matrix A into an upper triangular matrix, retaining integer entries:

$$a_{1,1}c_1 + a_{1,2}c_2 + \ldots + a_{1,\sigma}c_\sigma = m_1 \tag{22}$$

$$0c_1 + (a_{2,1}a_{1,2} - a_{1,1}a_{2,2})c_2 + \ldots + (a_{2,1}a_{1,\sigma} - a_{1,1}a_{2,\sigma})c_\sigma = a_{2,1}m_1 - a_{1,2}m_2$$

We have: $a_{2,1}a_{1,i} - a_{1,1}a_{2,i} \geq 0 \iff a_{2,1}a_{1,i} \geq a_{1,1}a_{2,i} \iff \frac{a_{1,i}}{a_{2,i}} \geq \frac{a_{1,1}}{a_{2,1}}$. We now construct ER tables for alphabet $\{a_{1,1}, \ldots, a_{1,\sigma}\}$ and for the new weighted alphabet $\{a_{2,1}a_{1,i} - a_{1,1}a_{2,i} \mid i = 2, \ldots, \sigma\}$. We then decompose $a_{2,1}m_1 - a_{1,2}m_2$ over this alphabet, checking in each step of the recursion in the first ER table whether the current solution is still feasible.

14 Computational results

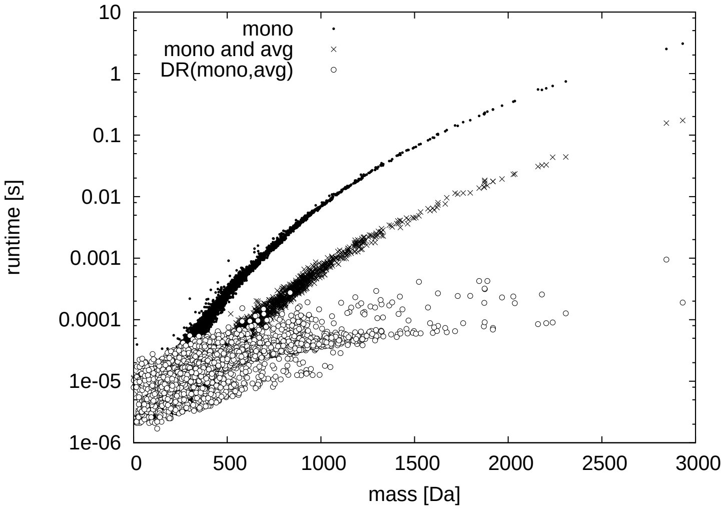

Fig. 3. Runtimes of decomposition algorithms in comparison (logarithmic scale): decomposing the monoisotopic mass with no additional information (dots), simultaneously decomposing the monoisotopic and average masses (crosses), and the DR-algorithm on monoisotopic and average masses (circles). Input data extracted from the KEGG LIGAND database.

As a first evaluation of our algorithms for decomposing metabolite isotope patterns, we decomposed molecular masses over the CHNOPS alphabet, using data from the KEGG LIGAND database [9]: We extracted 10 300 sum formulas over the alphabet CHNOPS, which reduced to 5 627 non-redundant sum formulas. We computed the monoisotopic and average masses and used these as input for our algorithms, using precision $\delta = 10^{-4}$ Da. Runtimes on a Sun Fire 880 with 900-MHz UltraSPARC-III-CPU, 32 GB RAM, are shown in Fig. 3: (i) computing all decompositions of the monoisotopic mass, (ii) doing the same respecting decompositions of the average mass of the molecule, and (iii) using the DR algorithm on the monoisotopic and average masses. Runtimes for $\delta = 10^{-3}$ Da are similar (data not shown).

Our experiments show that using dimension reduction as is done by the DR algorithm is greatly superior to using additive invariants directly. It should be noted, though, that these simulations are a proof of concept only, since for real applications, measurement errors need to be taken into account. This is dealt with in Part I, along with the ranking of solutions. It is realistic to expect that results will carry over straighforwardly, since the runtimes of all algorithms are effected in the same way.

15 Conclusion of Part II

We have studied the problem of decomposing isotope patterns, that is, computing the sum formula of an unknown molecule solely from its isotope pattern. In this context, we have presented methods for the efficient simulation of isotope patterns, as well as an approach to significantly reduce the search space of molecule candidates. We have shown that our algorithm for joint decompositions performs well on real data using the monoisotopic and the average mass of the molecule. In the future, we want to extend this approach by including the other additive invariants introduced in this paper, and by incorporating methods for respecting measurement errors (see Part I). Furthermore, we will test our method on validated isotope patterns of known metabolites.

Acknowledgments and References

Acknowledgments

AP supported by Deutsche Forschungsgemeinschaft (BO 1910/1), ZsL by Alexander von Humboldt Foundation and the Bundesministerium für Bildung und Forschung, within the group “Combinatorial Search Algorithms in Bioinformatics.” Additional programming by Marcel Martin, whom we thank for his unfailing support, and by Marco Kortkamp.

References

1. G. Audi, A. Wapstra, and C. Thibault. The AME2003 atomic mass evaluation (ii): Tables, graphs, and references. *Nucl. Phys. A*, 729:129–336, 2003.
2. M. Beck, I. M. Gessel, and T. Komatsu. The polynomial part of a restricted partition function related to the frobenius problem. *The Electronic Journal of Combinatorics*, 8(1):N7, 2001.
3. M. Bertrand, P. Thibault, M. Evans, and D. Zidarov. Determination of the empirical formula of peptides by fast atom bombardment mass spectrometry. *Biomed. Environ. Mass Spectrom.*, 14(6):249–256, 1987.
4. S. Böcker and Zs. Lipták. Efficient mass decomposition. In *Proc. of ACM Symposium on Applied Computing (ACM SAC 2005)*, pages 151–157, Santa Fe, USA, 2005.
5. S. Böcker and Zs. Lipták. The Money Changing Problem revisited: Computing the Frobenius number in time $O(k\,a_1)$. In *Proc. of Conf. on Computing and Combinatorics (COCOON 2005)*, volume 3595 of *Lect. Notes Comput. Sc.*, pages 965–974. Springer, 2005.
6. A. Fürst, J.-T. Clerc, and E. Pretsch. A computer program for the computation of the molecular formula. *Chemom. Intell. Lab. Syst.*, 5:329–334, 1989.
7. A. H. Grange, M. C. Zumwalt, and G. W. Sovocool. Determination of ion and neutral loss compositions and deconvolution of product ion mass spectra using an orthogonal acceleration time-of-flight mass spectrometer and an ion correlation program. *Rapid Commun Mass Spectrom*, 20(2):89–102, 2006.
8. C. S. Hsu. Diophantine approach to isotopic abundance calculations. *Anal. Chem.*, 56(8):1356–1361, 1984.
9. M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nuc. Acid Res.*, 34:D354–D357, 2006.
10. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, 2004.
11. T. Kind and O. Fiehn. Metabolomic database annotations via query of elemental compositions: Mass accuracy is insufficient even at less than 1 ppm. *BMC Bioinformatics*, 7(1):234, Apr 2006.
12. H. Kubinyi. Calculation of isotope distributions in mass spectrometry: A trivial solution for a non-trivial problem. *Anal. Chim. Acta*, 247:107–119, 1991.
13. Zs. Lipták. *Strings in Proteomics and Transcriptomics: Algorithmic and Combinatorial Questions in Mass Spectrometry and EST Clustering.* PhD thesis, Bielefeld University, Technical Faculty, 2005. Available from `http://bieson.ub.uni-bielefeld.de/frontdoor.php?source_opus=860`.
14. S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Chichester, 1990.

15. F. W. McLafferty. *Wiley Registry of Mass Spectral Data*. John Wiley & Sons, 7th edition with NIST 2005 spectral data edition, 2005.
16. S. Ojanperä, A. Pelander, M. Pelzing, I. Krebs, E. Vuori, and I. Ojanperä. Isotopic pattern and accurate mass determination in urine drug screening by liquid chromatography/time-of-flight mass spectrometry. *Rapid Commun Mass Spectrom*, 20(7):1161–1167, Mar 2006.
17. V. Pellegrin. Molecular formulas of organic compounds: the nitrogen rule and degree of unsaturation. *J. Chem. Educ.*, 60(8):626–633, 1983.
18. S. C. Pomerantz, J. A. Kowalak, and J. A. McCloskey. Determination of oligonucleotide composition from mass spectromectrically measured molecular weight. *J. Am. Soc. Mass Spectr.*, 4:204–209, 1993.
19. A. L. Rockwood and S. L. Van Orden. Ultrahigh-speed calculation of isotope distributions. *Anal. Chem.*, 68:2027–2030, 1996.
20. A. L. Rockwood, J. R. Van Orman, and D. V. Dearden. Isotopic compositions and accurate masses of single isotopic peaks. *J. Am. Soc. Mass Spectr.*, 15:12–21, 2004.
21. V. Shevchenko. *Qualitative Topics in Integer Linear Programming*. American Mathematical Society, 1996.
22. J. van Lint and R. Wilson. *A Course in Combinatorics*. Cambridge University Press, 2001.
23. M. Wehofsky, R. Hoffmann, M. Hubert, and B. Spengler. Isotopic deconvolution of matrix-assisted laser desorption/ionization mass spectra for substance-class specific analysis of complex samples. *Eur. J. Mass Spectrom.*, 7:39–46, 2001.
24. H. Wilf. *generatingfunctionology*. Academic Press, 1990.
25. J. A. Yergey. A general approach to calculating isotopic distributions for mass spectrometry. *Int. J. Mass Spectrom. Ion Phys.*, 52(2–3):337–349, 1983.
26. N. Zhang, R. Aebersold, and B. Schwikowski. ProbID: a probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, 2(10):1406–1412, Oct 2002.
27. W. Zhang and B. T. Chait. ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. *Anal. Chem.*, 72(11):2482–2489, 2000.

Bisher erschienene Reports an der Technischen Fakultät
Stand: 2007-05-14

94-01 Modular Properties of Composable Term Rewriting Systems
(Enno Ohlebusch)

94-02 Analysis and Applications of the Direct Cascade Architecture
(Enno Littmann, Helge Ritter)

94-03 From Ukkonen to McCreight and Weiner: A Unifying View of Linear-Time Suffix Tree Construction
(Robert Giegerich, Stefan Kurtz)

94-04 Die Verwendung unscharfer Maße zur Korrespondenzanalyse in Stereo Farbbildern
(André Wolfram, Alois Knoll)

94-05 Searching Correspondences in Colour Stereo Images – Recent Results Using the Fuzzy Integral
(André Wolfram, Alois Knoll)

94-06 A Basic Semantics for Computer Arithmetic
(Markus Freericks, A. Fauth, Alois Knoll)

94-07 Reverse Restructuring: Another Method of Solving Algebraic Equations
(Bernd Bütow, Stephan Thesing)

95-01 PaNaMa User Manual V1.3
(Bernd Bütow, Stephan Thesing)

95-02 Computer Based Training-Software: ein interaktiver Sequenzierkurs
(Frank Meier, Garrit Skrock, Robert Giegerich)

95-03 Fundamental Algorithms for a Declarative Pattern Matching System
(Stefan Kurtz)

95-04 On the Equivalence of E-Pattern Languages
(Enno Ohlebusch, Esko Ukkonen)

96-01 Static and Dynamic Filtering Methods for Approximate String Matching
(Robert Giegerich, Frank Hischke, Stefan Kurtz, Enno Ohlebusch)

96-02 Instructing Cooperating Assembly Robots through Situated Dialogues in Natural Language
(Alois Knoll, Bernd Hildebrand, Jianwei Zhang)

96-03 Correctness in System Engineering
(Peter Ladkin)

96-04 An Algebraic Approach to General Boolean Constraint Problems
(Hans-Werner Güsgen, Peter Ladkin)

96-05 Future University Computing Resources
(Peter Ladkin)

96-06 Lazy Cache Implements Complete Cache
(Peter Ladkin)

96-07 Formal but Lively Buffers in TLA+
(Peter Ladkin)

96-08 The X-31 and A320 Warsaw Crashes: Whodunnit?
(Peter Ladkin)

96-09 Reasons and Causes
(Peter Ladkin)

96-10 Comments on Confusing Conversation at Cali
(Dafydd Gibbon, Peter Ladkin)

96-11 On Needing Models
(Peter Ladkin)

96-12 Formalism Helps in Describing Accidents
(Peter Ladkin)

96-13 Explaining Failure with Tense Logic
(Peter Ladkin)

96-14 Some Dubious Theses in the Tense Logic of Accidents
(Peter Ladkin)

96-15 A Note on a Note on a Lemma of Ladkin
(Peter Ladkin)

96-16 News and Comment on the AeroPeru B757 Accident
(Peter Ladkin)

97-01 Analysing the Cali Accident With a WB-Graph
(Peter Ladkin)

97-02 Divide-and-Conquer Multiple Sequence Alignment
(Jens Stoye)

97-03 A System for the Content-Based Retrieval of Textual and Non-Textual
Documents Based on Natural Language Queries
(Alois Knoll, Ingo Glöckner, Hermann Helbig, Sven Hartrumpf)

97-04 Rose: Generating Sequence Families
(Jens Stoye, Dirk Evers, Folker Meyer)

97-05 Fuzzy Quantifiers for Processing Natural Language Queries in Content-Based Multimedia Retrieval Systems
(Ingo Glöckner, Alois Knoll)

97-06 DFS – An Axiomatic Approach to Fuzzy Quantification
(Ingo Glöckner)

98-01 Kognitive Aspekte bei der Realisierung eines robusten Robotersystems für Konstruktionsaufgaben
(Alois Knoll, Bernd Hildebrandt)

98-02 A Declarative Approach to the Development of Dynamic Programming Algorithms, applied to RNA Folding
(Robert Giegerich)

98-03 Reducing the Space Requirement of Suffix Trees
(Stefan Kurtz)

99-01 Entscheidungskalküle
(Axel Saalbach, Christian Lange, Sascha Wendt, Mathias Katzer, Guillaume Dubois, Michael Höhl, Oliver Kuhn, Sven Wachsmuth, Gerhard Sagerer)

99-02 Transforming Conditional Rewrite Systems with Extra Variables into Unconditional Systems
(Enno Ohlebusch)

99-03 A Framework for Evaluating Approaches to Fuzzy Quantification
(Ingo Glöckner)

99-04 Towards Evaluation of Docking Hypotheses using elastic Matching
(Steffen Neumann, Stefan Posch, Gerhard Sagerer)

99-05 A Systematic Approach to Dynamic Programming in Bioinformatics. Part 1 and 2: Sequence Comparison and RNA Folding
(Robert Giegerich)

99-06 Autonomie für situierte Robotersysteme – Stand und Entwicklungslinien
(Alois Knoll)

2000-01 Advances in DFS Theory
(Ingo Glöckner)

2000-02 A Broad Class of DFS Models
(Ingo Glöckner)

2000-03 An Axiomatic Theory of Fuzzy Quantifiers in Natural Languages
(Ingo Glöckner)

2000-04 Affix Trees
(Jens Stoye)

2000-05 Computergestützte Auswertung von Spektren organischer Verbindungen
(Annika Büscher, Michaela Hohenner, Sascha Wendt, Markus Wiesecke, Frank Zöllner, Arne Wegener, Frank Bettenworth, Thorsten Twellmann, Jan Kleinlützum, Mathias Katzer, Sven Wachsmuth, Gerhard Sagerer)

2000-06 The Syntax and Semantics of a Language for Describing Complex Patterns in Biological Sequences
(Dirk Strothmann, Stefan Kurtz, Stefan Gräf, Gerhard Steger)

2000-07 Systematic Dynamic Programming in Bioinformatics (ISMB 2000 Tutorial Notes)
(Dirk J. Evers, Robert Giegerich)

2000-08 Difficulties when Aligning Structure Based RNAs with the Standard Edit Distance Method
(Christian Büschking)

2001-01 Standard Models of Fuzzy Quantification
(Ingo Glöckner)

2001-02 Causal System Analysis
(Peter B. Ladkin)

2001-03 A Rotamer Library for Protein-Protein Docking Using Energy Calculations and Statistics
(Kerstin Koch, Frank Zöllner, Gerhard Sagerer)

2001-04 Eine asynchrone Implementierung eines Microprozessors auf einem FPGA
(Marco Balke, Thomas Dettbarn, Robert Homann, Sebastian Jaenicke, Tim Köhler, Henning Mersch, Holger Weiss)

2001-05 Hierarchical Termination Revisited
(Enno Ohlebusch)

2002-01 Persistent Objects with O2DBI
(Jörn Clausen)

2002-02 Simulation von Phasenübergängen in Proteinmonoschichten
(Johanna Alichniewicz, Gabriele Holzschneider, Morris Michael, Ulf Schiller, Jan Stallkamp)

2002-03 Lecture Notes on Algebraic Dynamic Programming 2002
(Robert Giegerich)

2002-04 Side chain flexibility for 1:n protein-protein docking
(Kerstin Koch, Steffen Neumann, Frank Zöllner, Gerhard Sagerer)

2002-05 ElMaR: A Protein Docking System using Flexibility Information
(Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, Gerhard Sagerer)

2002-06 Calculating Residue Flexibility Information from Statistics and Energy based Prediction
(Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, Gerhard Sagerer)

2002-07 Fundamentals of Fuzzy Quantification: Plausible Models, Constructive Principles, and Efficient Implementation
(Ingo Glöckner)

2002-08 Branching of Fuzzy Quantifiers and Multiple Variable Binding: An Extension of DFS Theory
(Ingo Glöckner)

2003-01 On the Similarity of Sets of Permutations and its Applications to Genome Comparison
(Anne Bergeron, Jens Stoye)

2003-02 SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry
(Sebastian Böcker)

2003-03 From RNA Folding to Thermodynamic Matching, including Pseudoknots
(Robert Giegerich, Jens Reeder)

2003-04 Sequencing from compomers: Using mass spectrometry for DNA de-novo sequencing of 200+ nt
(Sebastian Böcker)

2003-05 Systematic Investigation of Jumping Alignments
(Constantin Bannert)

2003-06 Suffix Tree Construction and Storage with Limited Main Memory
(Klaus-Bernd Schürmann, Jens Stoye)

2003-07 Sequencing from compomers in thepresence of false negative peaks
(Sebastian Böcker)

2003-08 Genalyzer: An Interactive Visualisation Tool for Large-Scale Sequence Matching – Biological Applications and User Manual
(Jomuna V. Choudhuri, Chris Schleiermacher)

2004-01 Sequencing From Compomers is NP-hard
(Sebastian Böcker)

2004-02 The Money Changing Problem revisited: Computing the Frobenius number in time

$O(k\,a_1)$
(Sebastian Böcker, Zsuzsanna Lipták)

2004-03 Accelerating the Evaluation of Profile HMMs by Pruning Techniques
(Thomas Plötz, Gernot A. Fink)

2004-04 Optimal Group Testing Strategies with Interval Queries and Their Application to Splice Site Detection
(Ferdinando Cicalese, Peter Damaschke, Ugo Vaccaro)

2004-05 Compressed Representation of Sequences and Full-Text Indexes
(Paolo Ferragina, Giovanni Manzini, Veli Mäkinen, Gonzalo Navarro)

2005-01 Overlaps Help: Improved Bounds for Group Testing with Interval Queries
(Ferdinando Cicalese, Peter Damaschke, Libertad Tansini, Sören Werth)

2005-02 Two batch Fault-tolerant search with error cost constraints: An application to learning
(Ferdinando Cicalese)

2005-03 Searching for the Shortest Common Supersequence
(Sergio A. de Carvalho Jr., Sven Rahmann)

2005-04 Counting Suffix Arrays and Strings
(Klaus-Bernd Schürmann, Jens Stoye)

2005-05 Alignment of Tandem Repeats with Excision, Duplication, Substitution and Indels (EDSI)
(Michael Sammeth, Jens Stoye)

2005-06 Statistics of Cleavage Fragments in Random Weighted Strings
(Hans-Michael Kaltenbach, Henner Sudek, Sebastian Böcker, Sven Rahmann)

2006-01 Decomposing metabolomic isotope patterns
(Sebastian Böcker, Zsuzsanna Lipták, Anton Pervukhin)

2006-02 On Common Intervals with Errors
(Cedric Chauve, Yoan Diekmann, Steffen Heber, Julia Mixtacki, Sven Rahmann, Jens Stoye)