13 Metabolite Mass Spectrometry

> *"A mathematician is a device for turning caffeine into theorems."* (Alfréd Rényi, modified)

THE phenotype of an organism is strongly determined by the small chemical compounds contained in its cells. These compounds are called *metabolites*; their mass is typically below 1000 Da. Metabolites are the intermediates and products of metabolism, that is, chemical reactions that happen in living beings to maintain life. Biopolymers such as proteins, DNA, or glycans (see Chapter 14 below) are not considered metabolites, but their constituent monomers (amino acids, monosaccharides) are. Small biopolymers such as peptides with only two amino acids (for example, carnosine) or disaccharides (for example, sucrose) are often also considered metabolites.

Metabolites can be subdivided into two major classes. A *primary* metabolite is directly involved in growth, development, and reproduction of a cell or organism: For example, adenosine-5'-triphosphate (ATP) is the energy currency of the cell. A secondary metabolite is not directly involved in those processes. Examples include antibiotics and pigments; a secondary metabolite of particular importance to science in general, is shown in Fig. 13.1.

A major challenge is that most of the secondary metabolites in any given higher eukaryote are largely unknown: Current estimates are in the range of up to 20 000 metabolites for any given species. In particular, plants, filamentous fungi, and marine bacteria synthesize enormous numbers of secondary metabolites. Unlike for proteins, genome sequencing usually does not allow us to deduce the structure of the metabolites.

Another challenge that we have to face, it that the chemical structure of metabolites is not restricted: Unlike for biopolymers who are made from smaller monomer building blocks in some ordered fashion (strings for proteins, trees for glycans) the molecular structure of metabolites

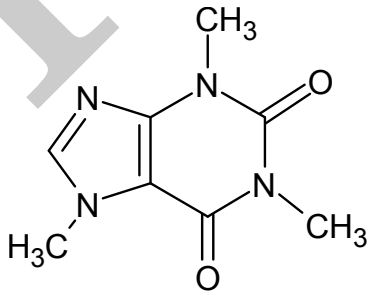

Figure 13.1: Left: Structural formula of caffeine, molecular formula $C_8H_{10}N_4O_2$, monoisotopic mass 194.080376 Da. Right: Corresponding molecular graph with colors from $\{C, H, N, O\}$. Note that some hydrogen atoms are omitted in the structural formula. **[TODO: REDRAW STRUCTURAL FORMULA, DRAW GRAPH WITH VERTICES $1, 2, \ldots$.]**

is not restricted. In spite of the small size of metabolites, this results in a huge variety and complexity of such molecules.

For the analysis via mass spectrometry, the important point is not that metabolites are intermediates or products of some metabolism; the important point is that they are *small molecules*: These are molecules with mass below about 1000 Da that, by definition, are no biopolymers. See above for exceptions to the "no polymer" rule. It is understood that there exist small molecules that are not metabolites: To name two examples, consider antiviral drugs and pesticides, which are often "artificial".

Much like for proteins, mass spectrometry is also a key technology for the identification of small molecules. Various analytical setups have been developed, most notably gas chromatography MS (GC-MS, also GC/MS) and liquid chromatography MS (LC-MS, see Chapter 11). GC-MS is almost exclusively with Electron Ionization (EI), that both ionizes and fragments compounds. GC-MS spectra can be interpreted automatically via database search, since reference spectra were collected over many years and the fragmentation mechanisms are reproducible on different instruments. Large spectral libraries of measured GC-MS reference spectra are available, such as the commercial NIST library (Gaithersburg, USA).

GC-MS requires the metabolite to be thermally stable. Unfortunately, this is not the case for several biologically important compound classes: Just imagine what happens to sugar-containing metabolites when you heat them. These compounds can be analyzed using LC-MS. Recall the experimental setup from Chapter 11: First, compounds are separated by liquid chromatography, and MS^1 spectra are recorded; the parent ion of one compound is mass selected, and fragmented by collision-induced dissociation; and masses of resulting fragments are recorded in a tandem mass spectrum. Compared to GC-MS, this has the additional advantage that we can fragment one compound at a time, whereas fragmentation by EI is applied to all compounds that leave the GC column simultaneously. The computational analysis of metabolite tandem MS data is still in its infancy, and this is presumed to be one of the major technological hurdles in metabolomics today. The time-consuming manual analysis of these data is actually non-trivial, as the fragmentation of small molecules under varying fragmentation energies is not completely understood. Due to the limited reproducibility of CID mass spectra on different instruments, even searching in spectral libraries is a serious problem.

This chapter is somewhat different from the previous ones, in that the computational analysis of metabolite MS is still in its infancy. This is somewhat surprising, as first attempts have been made in back in 1965 (see Sec. 13.6), one year before tandem mass spectrometry was invented. Tens or even hundreds of papers have been published on the topic, but even 35 years later, not much progress has been made regarding *general* methods for analyzing such data. To this end, I will first describe rather shallowly two "classical" approaches for interpreting such data, see Sec. 13.1 and 13.2. Then, we come back to combinatorial optimization, which will pave a completely new way towards analyzing such data in Sec. 13.4. Finally, in Sec. 13.6, I say a few words on the DENDRAL project already mentioned in the foreword.

13.1 Rule-based approaches for predicting and interpreting fragmentation

The classical way to approach the interpretation of a fragmentation mass spectrum, is to learn as much as possible about the fragmentation processes, and then to apply your knowledge to the data at hand. In fact, a lot is known, in particular about fragmentation by Electron Ionization:

For example, the complete Chapter 6 (more than 100 pages) of the textbook by Gross [102] is devoted to the fragmentation through EI. Here, we can take three different roads: Either, we try to interpret the mass spectrum using our knowledge about fragmentation; or, we have a hypothesis about the molecular structure that we are analyzing, and we try to predict the fragmentation spectrum using this hypothesis. Finally, we can combine both approaches, what will be covered in the next section.

Automating these approaches into rule-based computer programs or "expert systems", is definitely not a new idea: Back in 1969, Venkataraghavan *et al.* [228] presented an automated approach "to identify the general nature of the compound and its functional groups." Later, STIRS [139] mixes this rule-based approach with some early machine learning techniques. On a related note, see Sec. 13.6 below for information on the DENDRAL project, started back in 1965.

[TODO: BESCHREIBEN: GASTEIGER *et al.* [88]]

We noted in the introduction of this chapter that — quite different from EI fragmentation — the fragmentation of small molecules by tandem mass spectrometry is not completely understood, in particular under varying fragmentation energies. This is also much unlike glycan fragmentation and, even more so, peptide fragmentation: There, not only the "basic fragmentation concepts" have been identified but also, involved chemical models for fragmentation reactions (in particular, rearrangements) have been developed over the past years.

Two commercial programs are available for rule-based prediction and interpretation of fragmentation data: These are Mass Frontier and ACD/MS Fragmenter. Mass Frontier contains about 100 000 reactions collected from mass spectrometry literature, but this number is ever-increasing. It can both explain a measured fragmentation spectrum, and predict a spectrum from a molecular structure. Mass Frontier was developed by HighChem, Ltd. (Bratislava, Slovakia), but versions after 5.0 are available from Thermo Scientific (Waltham, USA). ACD/MS Fragmenter is targeted at predicting a fragmentation spectrum, given that the molecular structure of some compound is known. ACD/MS Fragmenter is available from Advanced Chemistry Labs (Toronto, Canada). Both programs were initially tailored towards the prediction and interpretation of fragmentation by EI, but recently, there has been a tendency to interpret tandem MS data with theses programs. Clearly, not too many details have been published about either approach, with the natural exceptions of articles claiming that the methods work for this or that application.

While rule based-approaches may be successful for certain areas of chemoinformatics [40] (and the interpretation of small molecules is as much chemoinformatics as it is bioinformatics), such approaches have not had much impact on the interpretation of small molecule MS data; compare this to the (almost) fully automated analysis used in shotgun proteomics pipelines. I do not want to speculate about the reasons for this failure, but I believe that many of them are inherently tied to rule-based decision making: Even commercial systems such as Mass Frontier with its 100 000 reactions, cover only a tiny part of the rules that should be known; be it that other rules cannot be learned from the currently known "chemistry universe", or that nobody found them by now. The number of rules is ever-increasing, possibly at an exponential rate; in the worst case, the number of rules that you *should* know, may be infinite. Rule-based systems usually do not scale well when more rules are added, so heuristics are used to trim the search space what, in turn, might make it impossible to find the correct answer. As Chen [40] notes, "it turned out that expert systems with real-problem solving capability were more difficult to build, to maintain, and to use than one had expected."

But none of the "big problems" in bioinformatics (such as sequence alignment, genome assembly, or reconstructing phylogenetic trees, to name just a few) has ever been solved by

an expert-curated rule-based system. Most problems are solved by combinatorial optimization, maximizing some objective function. Even certain Machine Learning methods, such as Support Vector Machines, follow this mantra. Chen [40] compares chemoinformatics expert systems with IBM's Deep Blue, defeating Garry Kasparov in 1997, but misses the main point: The reason why Deep Blue plays so well is that its core is *combinatorial optimization*, whereas rule-based approaches are only used for, say, the opening. Even an excellent opening is useless, if you are clueless after move ten. Cobbler, stick to your trade: We will now get back to combinatorial optimization, and see where that gets us. After one has sorted out the basics of combinatorial optimization, expert knowledge and rule-based decisions can be integrated into such approaches via, say, score modifications; compare to Chapter 8.

13.2 The somewhat omniscient oracle: Matching molecular structure and experimental data

Let us assume for a moment that we have both the experimental tandem MS data, *and* the correct molecular structure of the compound: All we have to do, is match what we see experimentally, with what we know is true. You might wonder that this is a little bit too much information at the same time. But as CID fragmentation of metabolites is still largely not understood, that is actually a quite sensible question.

Obviously, this problem can again be approached by rule-based methods, and this has been done as part of the DENDRAL project, see Sec. 13.6. But this approach has not been tremendously successful, at least with regards to the automated structure elucidation; and as mentioned at the end of the previous section, there are good reasons for that. So, we will take a short look at two optimization-based approaches for the problem.

Unfortunately, many negative results are weighting along the path to this problem. An obvious sub-problem is, given a molecular structure, is there a substructure of mass m? To make things somewhat simpler for us, let us assume that we have successfully transformed the mass of the fragment into its molecular formula. Then, the question is: Is there a substructure of the molecule with the given molecular formula? We now formalize this problem: Assume that we are given a colored, undirected graph $G = (V, E)$: The molecular structure is encoded in the graph, whereas elements are encoded in the colors, see Fig. 13.1. A molecular formula corresponds to a *multiset* of colors: A multiset is a set where, for each element in the set, we also record the number of times it appears. Now, our question is: Is there an induced subgraph G' of the given graph, such that the multiset of colors of this subgraph equals a given input multiset of colors?

In theoretical computer science, this problem is known as the GRAPH MOTIF problem. In 2006, Lacroix *et al*. [140] proposed this problem for searching motifs in metabolomic networks; since then, a series of theoretical papers has proven the hardness of the problem with respect to different algorithmic flavors. On the positive side, several parameterized algorithms were constructed. We just note that the problem is NP-hard even for bipartite graphs of maximum vertex degree four and only two colors. Consequently, the bioinformatics community has mostly abandoned the problem in favor of other, better tractable formulations of network motifs.

For the application that we have in mind, the GRAPH MOTIF problem is an oversimplification, as we ignore the cost of cutting out a fragment at a particular position; incorporating edge weights usually results in increased running time. More importantly, this is only a tiny sub-

problem of the problem that we truly want to address; namely, explain all of the observed peaks simultaneously. We will come back to the second point later.

But somewhat counterintuitively, the problem that we address here, is *simpler* than the GRAPH MOTIF problem: As the fragmentation energy is limited, the molecule cannot break at too many positions simultaneously. So, we assume that we are given an upper bound b, such that at most b edge removals must suffice to disconnect the subgraph G from the remainder of G. In application, we may assume that this b is rather small, such as $b = 3$. Assume that we have some objective function $w : E \to \mathbb{R}$ that we want to minimize; w may correspond to the energy required to break some bond. Now, we ask: What is the induced subgraph G' of G that can be separated from the remainder of G via deleting an cut set $E' \subseteq E$ with $|E'| \leq b$, such that $w(E')$ is minimum? As we require $|E'| \leq b$, there may be no such graph.

We can try all edge sets with at most b edges, and see if they produce the desired multiset (molecular formula). As there are $O(|E|^b)$ such edge sets, an algorithm for that purpose will run in roughly that time. But we want to something slightly smarter: We use a branch-and-bound heuristic, aborting our search as soon as no optimal solution can be found in the future. Given a cut set $E' \subseteq E$ with $|E'| \leq b$, its deletion might separate G into a set of connected components. We use depth-first search to identify the connected components in G; simultaneously, colors are counted and costs for the partition are calculated. If the colors of a connected component correspond to the colors of the input multiset, and the costs are smaller than the costs of the best solution found so far, the connected component is stored. These costs w^* are then used as an upper bound for pruning.

Initially, edges are sorted in increasing order with respect to their weight. We use edge set iterators $i_1 < i_2 < \cdots < i_b$ pointing to the sorted set E. We iterate $i_1 = 1, \ldots, |E|$, and inside this loop iterate $i_2 = i_1 + 1, \ldots, |E|$, and so on, compare to Algorithm 10.1 on page 113. Let $w(i)$ be the cost of the edge that corresponds to iterator i. Assume that we have iterators $i_1, \ldots, i_{a-1}$ fixed, and that we want to iterate $i_a = i_{a-1} + 1, \ldots, |E|$ for $a \leq b$. Assume further that the weight of the partial solution for iterators $i_1, \ldots, i_{a-1}$ is known; we compute the weight of the partial solution for iterators $i_1, \ldots, i_a$ in constant time, adding $w(i_a)$, and pass this weight to the following iteration steps. We abort this loop if our current weight exceeds the current minimum weight w^*.

Another trick of speeding up this algorithm in applications, is to iterate over $b' = 1, \ldots, b$, and to examine only cut sets E' with $|E'| = b'$. Note that you may not stop if you found a solution for some $b' < b$, as cut sets with larger cardinality may have smaller weight. Another trick is to provide not only b as the maximum number of edges we are allowed to break, but also some maximum weight w^* that we are allowed to use for breaking edges.

Analyzing the worst-case running time of our algorithm is quite simple, as none of the heuristic improvements discussed above, does anything good to the worst-case running time. Sorting edges costs $O(|E| \log |E|)$ time. Running time of the depth first search is $O(|V|)$. The branch-and-bound algorithm iterates over $O(|E|^b)$ edge sets. This results in an overall running time of $O(|E| \log |E| + |V| \cdot |E|^b)$. It is easy to see that this algorithm also answers the question, whether there is a substructure of some given mass m.

Now, let us come back to the problem that we originally wanted to address: Given a tandem mass spectrum and the correct molecular structure of the compound; how can we explain the spectrum, given the structure?

[TODO: PASS OP!]

13.3 Searching in molecular structure databases

Interpreting experimental data when you already know what is in there, might be interesting for MS experts; it is definitely not that interesting for the general public. The interesting point is, that the methods from Sec. 13.2, but also the rule-based prediction of mass spectra (see below), can make a second career in database searching: Assume that we have measured tandem mass spectra of a compound that we want to identify. In addition, we have access to a large database of molecular structures, such as PubChem. We filter all compounds from the database that have the correct parent mass; these are our candidates. For every candidate molecular structure, we apply the above matching procedure; we ignore the actual matching, and only record the score of the matching. If our scoring is done in a sensible way, then the best matching between the experimental MS data and a molecular structure will also receive the highest score. We will call this approach *spectrum-structure matching* (SSM).

So, we have replaced database searching (in spectral libraries) by database searching (in molecular structure databases) — why bother?. In fact, this has gained us a lot: Spectral libraries are (and will be) tiny, in comparison to the huge molecular structure databases. The CAS Registry of the American Chemical Society and PubChem currently contain about 25 million compounds each. The actual numbers are of no importance here; it is enough to say that there are "a lot" of compounds. In comparison, spectral libraries unusually contain at most thousands of compounds and, hence, are several orders of magnitude smaller.

In principle, we can also combine SSM with structure generators such as MOLGEN [14, 131, 132]. Structure generators enumerate, for a given molecular formula or mass or some similar input, all molecular structures that are chemically sound. This allows us to overcome the boundaries of database searching: Simply generate all molecular structures corresponding to the parent mass or molecular formula, and use the output of the structure generator as our private database, compare to Sec. 8.4. But unfortunately, there is little hope that this approach will ever work: The problem is not the structure generators, that can enumerate millions of structures in a matter of seconds. The problem is the size of the search space: For example, MOLGEN enumerates more than 100 million molecular structures for the molecular formula $C_8H_6N_2O$ with mass 146.048013, in a matter of minutes [132]. Now, ranking these structures by SSM, to find the one that is correct, is an extremely hard task. In fact, it turns out that SSM is already somewhat overstrained with the hundreds of molecular structures, that we find in molecular structure databases for a particular parent mass [235]. Except for tiny compounds, there is little hope to push SSM into the realms of *in silico* structure generators.

A similar approach can be used for rule-based prediction of mass spectra: Here, we query the database for all molecular structures that have the correct mass. For each molecular structure, we use a rule-based approach to compute a theoretical fragmentation mass spectrum. Finally, we compare the simulated fragmentation spectra to the measured spectrum, and ranked compounds by the peak counting score, or a similar score for spectra comparison. Note that this approach withholds the measured spectrum information from the rule-based method. In all likelihood, this will make it much harder to come up with the correct rules. So, it appears advisable to to use the approach based on the "somewhat omniscient oracle" instead.

13.4 True *de novo* interpretation via fragmentation trees

We noted that small compounds can fragment at almost any chemical bond, and that the fragmentation process is not completely understood and difficult to predict. We now account for this missing comprehension, by allowing *arbitrary* fragmentation.

In the following, we are analyzing the tandem MS data of a single small compound. We assume that the molecular formula of the compound is known: This can be achieved using isotope pattern analysis (see Chapter 10), or by a modification of the method presented here, see Sec. 13.7. To increase the amount of information available to us, we demand that several tandem mass spectra of the unknown compound are measured at different collision energies: Higher energies lead to smaller fragments, as more chemical bonds break. For ease of presentation, we merge these spectra into one, also merging peaks from different spectra with masses "sufficiently close." Alternatively, one tandem mass spectrum can been measured in "ramp mode", where the collision energy is varied while measuring the spectrum. Small compounds, and fragments thereof, usually carry a single charge, so we do not have to correct for different charge states. Most compounds and fragments are charged by a single proton, whereas others carry an intrinsic charge, for example, if quaternary nitrogen is present in the compound. To this end, we can add an electron mass to all peak masses, resulting in the masses of the uncharged compound and fragments.

In experiments, we see that fragments of the parent molecule are fragmented for a second time when higher collision energies are applied, compare to Sec. 13.2. In fact, ion trap mass spectrometers allows us to build such fragmentation cascades experimentally. To account for this multi-step fragmentation, we use a *fragmentation tree* to describe the fragmentation of the parent molecule: This tree has a set of molecular formulas as its vertex set, including the parent molecular formula. For each vertex molecular formula, there is a peak in the tandem MS data with mass withing measuring accuracy of the molecular formula mass: So, the vertices of the fragmentation tree *explain* the peak masses, or a subset thereof. Vertices are connected by directed edges, constituting *neutral losses*: That is parts of the molecule break off but are not ionized, so that we cannot detect them in the subsequent MS step. An directed edge (u,v) in the graph tells us that fragment v is a sub-molecule of fragment (or parent molecule) u: So, for each element, the molecular formula of v contains at most as many atoms as the candidate molecular formula of v.

Different fragmentation pathways may lead to fragments with identical molecular formula or even identical structure. Hence, we have slightly oversimplified the problem: By restricting ourselves to fragmentation *trees*, we demand that each fragment in the fragmentation spectrum is generated by a single fragmentation pathway. But this is not a serious oversimplification: Firstly, this situation is rather the exception than the rule. Secondly, we can argue that we are interested in the *major* fragmentation events that mainly occurred.

To compute a fragmentation tree, we decompose all fragment peak masses using Algorithm 10.2 on page 115. We only generate those molecular formulas which are sub-molecules of the candidate molecular formula, see Exercise 10.2. These molecular formulas, plus the parent molecular formula, are vertices of a directed graph named *fragmentation graph*. This graph is vertex-colored: Every vertex gets a color that represents the peak it explains, since one peak mass may be explained by a multitude of molecular formulas. We create a directed edge between two vertices if one molecular formula is a sub-molecule of the other molecular formula. Doing so, we represent every possible fragmentation step. Since the "sub-molecule" relation is transitive, the constructed graph is also transitive: $(u,v),(v,w) \in E$ implies $(u,w) \in E$.

To formulate our task as an optimization problem, our the first idea is, as usual, to use a peak counting score: We count the number of peaks that can be explained as fragments of the parent molecule. But here, this will not take us very far: Counting peaks, the fragmentation tree where every vertex is connected to the parent peak vertex, will receive maximum weight; in addition, there will be numerous other fragmentation trees with identical score. This implies that we will have to use a more involved scoring. For the ease of presentation, we sometimes assume unit weights, though.

So, let us calculate edge weights: Each vertex is assigned to a unique sum formula and a (usually non-unique) peak color, and we can score the vertex using properties such as peak intensity or mass deviation, see Sec. 4.2. It is theoretically unpleasant to score the trees using both weights on edges and vertices. To this end, we shift the vertex score to its incoming edge. This is possible as in the resulting tree, every vertex has exactly one incoming edge. The vertex score of the root can be ignored in our optimization, as it is part of all fragmentation trees. We use edge weights because we also want to score the hypothetical fragmentation step: for example, certain neutral losses are observed more frequently than others. Now, our task is to search the resulting graph for a subtree that has maximum weight. This is the fragmentation tree that we propose from the data.

One peak may result in many molecular formulas explaining it, and to avoid double counting of peaks, we have to ensure that the subtree does not use any color twice. Several fragments with different molecular formulas may result in a single peak in the fragmentation spectrum, but this is probably an extremely rare event in practice. To this end, we demand that our fragmentation tree is colorful: Each vertex color and, hence, each peak in the fragmentation spectrum is scored at most once.

We now formalize the problem of computing a fragmentation tree. All of the following definitions are standard in computational graph theory, see Sec. 17.2: A directed graph $T = (V,E)$ is a *tree*, if there is a root vertex $r \in V$ such that every vertex $v \in V$ can be reached from r via a unique path. Many alternative characterizations of trees exist; we just mention that T is a tree with root r if and only if every vertex $v \in V$ can be reached from r, and every vertex v but the root has in-degree one, whereas the root has in-degree zero. The fragmentation graph $G = (V,E)$ we are interested in, is a directed, acyclic graph (DAG). Hence, we will restrict our problem formulation to such graphs. Recall that *acyclic* means that we cannot walk away from some vertex v of the graph along directed edges, and ultimately end up in v again; in particular, trees are acyclic. Let $c : V \to C$ be the vertex colors (or peaks) of G. Furthermore, let $w : E \to \mathbb{R}$ be the edge weights, whose sum we want to maximize. A *subtree* $T = (V_T, E_T)$ of G is a subgraph of G, $V_T \subseteq V$ and $E_T \subseteq E$, that is a tree. The tree T is *colorful* if it uses every color in C at most once: so, $c(u) \neq c(v)$ holds for all $u, v \in V_T$ with $u \neq v$. We formalize our problem as:

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

Unfortunately, this problem is NP-hard, even if we assume that all edges have unit weight. As we know, this implies that we cannot compute a maximum colorful subtree in polynomial time, unless P = NP. To make the consequences clear: We cannot hope to find an algorithm with running time, say, $O(n^{1000})$, where n is the number of vertices of the graph, unless P = NP. In contrast, we can easily compute a maximum subtree if we drop the requirement that it has to be colorful, at least when all edges have non-negative weight. But it must be understood that limiting ourselves to colorful subtrees is inevitable, see Exercise 13.3. We mention in passing

two theoretical observation: Firstly, the MAXIMUM COLORFUL SUBTREE problem is a special case of the edge-weighted GRAPH MOTIF problem introduced in Sec. 13.2. This is somewhat surprising, as we were working with molecular graphs back then, whereas it is fragmentation graphs now. Second, the maximum subtree problem again becomes NP-hard, if we allow for negative edge weights in our DAG, see Exercise 13.4.

13.5 Algorithms for the Maximum Colorful Subtree problem

We will now look at three algorithms for the MAXIMUM COLORFUL SUBTREE problem: Whereas its computational hardness may be daunting at first, this does not mean that we have to abandon all hope! In the following, we assume that all colors C are in use, so $c(V) = C$ where $G = (V,E)$.

Our first algorithm is a heuristic, meaning that we cannot guarantee anything about the quality of the solution. Here, "quality of the solution" is obviously meant with regards to the objective function $w(T)$, as we cannot guarantee that the optimum tree is also the "true" tree. The heuristic that we want to employ, is well-known in computer science, and is called *greedy* heuristic: Given the fragmentation graph, we first sort the edges with respect to weight. We now add edges to the growing tree in the order of their edge weights, assuring at any point that we will end up with a colorful subtree.

During the course of computation, the resulting subgraph G' of $G = (V,E)$ is not necessarily connected. But we will make sure that at any time, G' is colorful, using every color of G at most once. Also, we will ensure that all vertices of G' have in-degree at most one, so that at a later stage, G' can be completed to a tree. Let G' be the subgraph we have constructed so far. Now, some edge $(u,v) \in E$ not in G' is *admissible*, if the graph that results from adding (u,v) to $G' = (V',E')$ respects these conditions: That is, $c(u) \neq c(w)$ holds for all $w \in V'$ with $w \neq u$; $c(v) \neq c(w)$ holds for all $w \in V'$ with $w \neq v$; and, the in-degree of v in G' is zero.

Now, the greedy algorithm can easily be described: Sort all edges $e \in E$ with respect to their edge weights $w(e)$. Initialize G' as the empty graph. Iterate over all edges $e \in E$, from heaviest to lightest. If e is admissible with respect to G', then add it to G'. Continue until all edges have been considered.

The above algorithm only works correctly, if there is only one source in G. If there are multiple sources in G, we iterate over all sources: For each source s, we consider separately the descent of s in G, which can be easily computed by a depth-first search in G. Then, we apply our algorithms to the resulting graph with a single source. Among the resulting tree, we accept that one with maximum weight.

What is the running time of our heuristic? We may assume that the graph G is connected; this implies $|V| \leq |E| + 1$, as every vertex of the graph must be incident to at least one edge, connecting it to the rest of the graph. Sorting E requires $O(|E| \log |E|)$ time. We keep an array of colors, allowing us to check in constant time if some color has been used before. With this, we can test whether some $e \in E$ is admissible, in constant time. So, the total running time for S sources is $O(S \cdot |E| \log |E|)$.

There are a few things that we want to point at: First, this heuristic computes a spanning tree, that spans all vertices of the input graph G. To improve the score of the subtree, we can clip off all leaves of the tree that are connected via an edge with negative weight. Second, this algorithm will compute the optimal solution, in case G is colorful itself, and all edge weights are non-negative. So, if the input graph G does not contain too many vertices with identical color, we expect the performance of the heuristic to be good.

The second algorithm is not much more complicated than the first. For this algorithm, we assume that all edge weights $w(e)$ are non-negative. Then, we have noted above that we can easily compute a maximum subtree, which is a spanning tree, in case $G = (V,E)$ is colorful: Assume that there is a single source s in G, so that all other vertices can be reached via directed paths. Then, a maximum spanning tree can be computed by choosing, for each vertex $v \in V$, the incoming edge with maximum weight. It is easy to check that the resulting graph is truly a tree. If there are multiple sources in G, we again iterate over all sources, as described above.

This algorithm has running time $O(S|E|)$ where S is the number of sources in G; we again assume G to be connected. In fact, it is easy to compute a maximum spanning tree, even if the directed graph is not a DAG, using the Chu-Liu/Edmonds Algorithm. Running time of this algorithm can be improved to $O(|E| + |V| \log |V|)$, but this is rather of theoretical interest, see below.

Unfortunately, our input graph is usually not colorful: What can we do? The simple answer is to make it colorful, by restricting it. For each color $c' \in C$, we choose one vertex $v \in V$ with $c(v) = c'$. We then look at the graph G' induced by this restricted vertex set $V' \subseteq V$, which is colorful. We compute a maximum spanning tree in G' by the above algorithm. We iterate over all subsets $V' \subseteq V$ with $c(V') = C$ and $c(u) \neq c(v)$ for $u, v \in V'$, $u \neq v$.

How many colorful subsets $V' \subseteq V$ exist? For a color $c' \in C$ let $n(c') := \#\{v \in V : c(v) = c'\}$ be the number of vertices in G of that color. It is easy to see that

$$cs := \#\text{colorful subsets} = \prod_{c' \in C} n(c').$$

Unfortunately, no useful bound or estimate of this number is possible, as it completely depends on color structure of the graph. Note that cs can get large even for relatively small graphs: For a graph with 100 vertices and ten colors, such that every color is used by ten vertices, we have $cs = 10^{10}$. We iterate over all colorful subsets, and reach a total running time of $O(cs \cdot S|E|)$. Alternatively, we can bound the running by $O(cs \cdot S|C|^2)$ as the restricted graph has $O(|C|)$ vertices and, hence, $O(|C|^2)$ edges. Note that we can compute the restricted graph in time $O(|C|^2)$ or $O(|E|)$, so this step of the computation is covered by our running time analysis.

[TODO: PASS OP!]

In addition to the problem size $|V|$, the number of vertices in the graph, we introduce a parameter $k := |C|$. This is the number of peaks in the spectrum, so typically $k \ll n$. We now present an algorithm that has superpolynomial running time in k, but polynomial running time in $|V|$. Such algorithms are called *parameterized* algorithms.

We can choose k arbitrarily in practice — heuristic to insert the remaining peaks in a greedy fashion.

We have noted at some places, that multiple sources in the DAG G require additional attention from our side. For computing fragmentation trees, multiple sources occur when we are not sure about the correct molecular formula of the parent ion, and want to base this decision on the fragmentation trees corresponding to the different molecular formulas. Some of the algorithms allow us to directly apply them even if multiple sources are present; others require us to iterate over the different sources. In practice, it is always a good idea to restrict the graph to a single source, iterating over its sources s. It turns out that all algorithms benefit from this restriction, in one way or the other. Considering the case of multiple sources, is only of interest from a theoretical perspective. For example, the DP algorithm seems to work as good for one source, as

it works for many. But if we restrict the graph to the descent of a single source s, this means that certain vertices and, possibly, certain colors are no longer present in the graph.

Which algorithm is best for what situation? If the mass accuracy of our measurement is very good, and if the compounds are not too large, then each peak in the mass spectrum will usually have one, sometimes two, but rarely more explanations as a molecular formula. In this case, both the greedy heuristic and the brute force approach are applicable: the former will then produce near-to-optimal results, the later will have acceptable running times and compute optimal solutions, at least if no negative edge weights are involved. But if the accuracy is getting worse, or compounds get larger, then many peaks in the spectrum will have several explanations. In this case, the quality of greedy solutions will deteriorate, and running times of the brute force approach will explode.

13.6 DENDRAL

First rule-based approaches for predicting fragmentation patterns, as well as explaining experimental mass spectra with the help of a molecular structure, were developed as part of the DENDRAL project that started back in 1965 [144, 145]: This lead to a series of papers, dealing with the interpretation of mass spectrometry data, and the identification of metabolites. Meta-DENDRAL was used to derived new "rules of thumb" for the analysis of mass spectrometry data, published in a series of papers in the *Journal of the American Chemical Society*. Gray *et al*. [98] describe the computer program CONGEN that implements a "somewhat omniscient oracle", see Sec. 13.2. Lavanchy *et al*. [143] use this method to interpret mass spectra of marine sterols. Gray *et al*. [99] do a similar analysis using the related GENOA program. See Mun and Mclafferty [169] and Smith *et al*. [215] for reviews of different aspects of the DENDRAL project at that time. Other parts of the project dealt with the automated analysis of Nuclear Magnetic Resonance (NMR) data [97]. A full coverage of all the techniques developed as part of the DENDRAL project, is far beyond the scope of this book; see Chapter 7 of the PhD thesis of November [174] for the early years of the project.[1]

[TODO: IN THE END NOT SUCCESSFUL.] Citing Gasteiger *et al*. [88]: "However, it is sad to say that, in the end, the DENDRAL project failed in its major objective of automatic structure elucidation by mass spectral data, and research was discontinued. Therefore, investigations of the relationships between structure and mass spectra by computer techniques suffered severe setbacks." Possibly, these people were too far ahead of their time: With high mass accuracy data and the compute power that we have today, things obviously become easier. Possibly, they chose the wrong objects to study: Peptides are much more convenient, from a computational perspective. As Biemann *et al*. [19] observed back in 1966, "the structures of oligopeptides follow a few strict requirements which can be simply expressed in computer language." But then, rule-based systems have not had much success in peptide analysis: There, it is apparent from the very beginning that, in view of the huge search space, only optimization- and combinatorics-based method can be successful. Finally, concepts from theoretical computer science, such as correctness proofs or worst-case running time analysis, did not play a role back then. Still, it might be a good idea for everybody who has a "new" idea on this topic, to look at "ye olde" publications.

[1] It is somewhat unfortunate that November [174] does not even touch upon the birth of bioinformatics, which was also at that time, and which indisputably had an impact on life science which surpasses that of DENDRAL by many orders of magnitude.

13.7 Historical notes and further reading

See Fernie *et al*. [78] and Last *et al*. [142] for introductions to metabolomics and metabolite profiling.

We claimed that genome sequencing does not allow us to deduce the structure of the metabolites. This is not true for polyketides, secondary metabolites that are made by polyketide synthases which, in turn, are huge proteins resembling conveyor belt factories.

Oberacher *et al*. [175, 176] present an approach for searching CID metabolite spectra in databases, that tackles the problem of low reproducibility of metabolite CID fragmentation.

Fellows *et al*. [72] show that the GRAPH MOTIF problem is NP-hard even for bipartite graphs of maximum vertex degree four and only two colors. Böcker *et al*. [30] evaluated weighted GRAPH MOTIF algorithms for cleaving fragments from a parent molecule, and found that the simple branch-and-bound heuristic presented here, performs best for this application. See Sikora [212] for a recent overview of theoretical results on the GRAPH MOTIF problem; and see Guillemot and Sikora [105] for some recent parameterized algorithm for different flavors of the problem, as well as increased running times when edge weights are taken into account.

To populate the edge weights of the molecule graph, we can use the enthalpy change upon bond fragmentation [154]; smarter ways of computing these weights would be beneficial, but might also lead to drastically increased running times.

The work of Heinonen *et al*. [109] is targeted at explaining what you see in a tandem mass spectrum of a metabolite, as introduced in Sec. 13.2. Their ILP-based approach suffers heavily from the complexity of the underlying problem, and the resulting combinatorial explosion; running times can be prohibitive even for medium-size molecules.

Hill *et al*. [110] implemented a "rule-based identification pipeline" as described in Sec. 13.1, searching PubChem as their molecular structure database, and using Mass Frontier 4 to predict tandem MS spectra from molecular structures. Wolf *et al*. [235] presented a somewhat greedy heuristic to match molecular structures and experimental data, see Sec. 13.3. Their approach slightly outperformed that of Hill *et al*. [110].

Our presentation of Sec. 13.4 largely follows the papers of Böcker and Rasche [26] and **?**]. Note that the approach from [26] was originally targeted only at finding the correct molecular formula of the compound. In this case, you compute an optimal fragmentation tree for every molecular formula that matches the parent mass; then, you rank these candidates with respect to the score of the computed optimal fragmentation tree. It turns out that by combining isotope pattern and fragmentation pattern data of an unknown compound, you can determine its molecular formula, without using databases or relying too much on "chemical knowledge" [**?**]. Böcker and Rasche [26] also describe how to compute edge weights of the fragmentation graph in a sensible way.

NP-hardness of the MAXIMUM COLORFUL SUBTREE problem was shown independently by Böcker and Rasche [26] and Fellows *et al*. [72]: In the latter paper, the more general GRAPH MOTIF problem is considered, which is then restricted to colorful motives and trees as input graphs, resulting in the unweighted MAXIMUM COLORFUL SUBTREE problem. In fact, the problem is also hard to approximate [59]: The problem has no constant factor approximation, unless P = NP.

The greedy heuristic presented in Sec. 13.5, is different from the one proposed in [**?**], and appears to perform much better in application; see [193]. The fast version of the Chu-Liu/Edmonds Algorithm [43, 65] is due to Gabow *et al*. [86].

?] show that **[TODO: PASS OP!]**.

Scheubert *et al*. [205] have modified the approach of calculating fragmentation trees, so that MS^n data can be taken into account. Interestingly, whereas it appears to be a much simpler task to reconstruct such trees from multiple MS data, the computational questions that arise in this context are even harder than those of Sec. 13.4.

For an introduction to parameterized algorithms, we refer the reader to [60, 173]. Using subsets of colors as part of the dynamic programming recurrence, has been used frequently in algorithmics: See for example Dreyfus and Wagner [61] who, back in 1972, colored graphs to compute a shortest Steiner tree.

13.8 Exercises

13.1 What happens to sugar-containing metabolites if you heat them? What do scientist do to prevent that, when they want to analyze metabolites by GC-MS?

13.2 Find all subgraphs with molecular formula CHN in the molecular graph of caffeine, see Fig. 13.1.

13.3 Why is it inevitable that we demand that the subtree of the fragmentation graph has to be colorful, when defining the MAXIMUM COLORFUL SUBTREE problem? Describe what happens if we drop this requirement.

13.4* Proof that finding a maximum subtree in a DAG that may contain negative edges, is an NP-hard problem.

14 Glycan *De Novo* Sequencing

"Sweets for my sweet — sugar for my honey!" (Doc Pomus and Mort Shuman)

GLYCANS are molecules made from simple sugars that form complex tree structures. Glycans constitute one of the most important Post-Translational Modifications of proteins: Apweiler *et al*. [4] estimate that more than 50 % of all eukaryotic proteins are glycosylated, i.e., carry a glycan modification. Glycans are believed to play an important role in cell growth and development, tumor growth and metastasis, immune recognition and response [190], and even the allergic reaction to white wine [181]. The elucidation of glycan structure remains one of the most challenging tasks in biochemistry. Like metabolites, but unlike proteins, the structure of glycans cannot be directly inferred from the genome sequence of an organism.

One of the most powerful tools for glycan structure elucidation is tandem mass spectrometry. As for peptide, glycan mass spectra can be interpreted by searching a database of glycan structures, but such databases are vastly incomplete.

In this chapter, we focus on the problem of *de novo* interpretation of glycan tandem MS data. This is very similar in spirit to peptide *de novo* sequencing, as introduced in Chapters 2 and 8. In fact, we will re-use several ideas from earlier chapters, and this chapter can be seen as an application of what we have already learned — with a twist. Note that the term "glycan sequencing" is somewhat ill-chosen, as glycans are trees and not sequences. We will stick with it, though, to avoid word monster such as "glycan *de novo* topology elucidation", and to underline the analogy to peptide *de novo* sequencing.

There are different levels of resolving the structure of a glycan: We concentrate on the "high-level" structure, namely, the "topology" of the glycan. In some sense, glycan sequencing is more difficult than peptide sequencing, as we try to resolve a tree structure (the topology of the glycan) instead of a linear string. We use a two-step approach as introduced in Sec. 8.4, the first step being *candidate generation* and the second step being *candidate evaluation*. As for peptides, we will focus on the candidate generation step. Many early tools for glycan sequencing use a naïve approach to generate candidates: They decompose the parent mass of the glycan over the alphabet of monosaccharides, then enumerate all topologies that have the correct multiplicities of monosaccharides. This approach faces the problem of a combinatorial explosion of structures. In the following, we will present a smarter way to generate candidates, based on the observed tandem MS data.

In Chapter 2 and Sec. 8.2, we have seen how to efficiently generate peptide candidates; our approach for generating glycan candidates will be closely related to that of Sec. 8.2. Recall that peptide sequencing can be solved in linear time if there is only one ion series, see Exercise 2.1. So, what is the twist here? It turns out that generating glycan candidates is computationally hard (NP-hard), even if we simultaneously restrict ourselves to (i) a single ion series, (ii) ideal data, and (iii) the peak counting score. This can be seen as a late justification for assuming ideal data in Chapter 2: Generating glycan candidates is computationally hard, and this hardness does not come from any peculiarities in the mass spectrometry data, but is an intrinsic part of

monosaccharides	mol. formula	mass (Da)
pentoses (Pen), such as xylose (Xyl)	$C_5H_8O_4$	132.040000
deoxyhexoses (dHex), such as fucose (Fuc)	$C_6H_{10}O_4$	146.06
hexoses (Hex), such as glucose (Glc), galactose (Gal), mannose (Man)	$C_6H_{12}O_6$	162.05
hexose acids (HexA), such as glucoronic acid (GlcA)	$C_6H_8O_6$	176.03
N-acetylhexosamines (HexNAc), such as N-acetylglucosamine (GlcNAc)	$C_8H_{13}NO_5$	203.08
N-acetylneuraminic acid (NeuAc, also Neu5Ac)	$C_{11}H_{17}NO_8$	291.10
N-glycolylneuraminic acid (NeuGc)	$C_{11}H_{17}NO_9$	307.09

Table 14.1: Monosaccharides commonly found in glycans, with molecular formula and monoisotopic mass of the residue (removed H_2O). **[TODO: CHECK FORMULAS, CALCULATE MASSES!]**

the combinatorial problem itself. If, in turn, we allow peaks to be counted multiple times, the resulting problem can be easily solved, but results can be pathetic, see for example Exercise 14.5.

We have seen in the previous section, that an NP-hard problem does not necessarily mean the end of all days. We will again derive a dynamic programming algorithm which does not only allow us to find the optimal solution, but also to sample suboptimal one. And again, the resulting algorithm is fixed-parameter tractable where, for our theoretical analysis, the parameter k is the "number of peaks in the measured spectrum". In practice, parameter k can be chosen arbitrarily and allows us to tune the methods, trading specificity for running time and memory consumption.

Finally, we want to prove our above claim that there truly is a combinatorial explosion of glycan topologies. To this end, we consider the problem of counting glycan topologies. In Sec. 14.7, we present methods for counting all glycan topologies with n monosaccharides, and for counting glycan topologies for a given mass m.

14.1 Glycans and glycan topologies

Glycans are — besides nucleic acids and proteins — the third major class of biopolymers, and are built from simple sugars (monosaccharides). Since monosaccharides can have up to five linkage sites, glycans can be assembled in a tree-like structure, making their primary structure considerably more complex than that of proteins. Glycans can be attached to proteins (N-glycans, O-glycans, glycosaminoglycans) or lipids, but may also be free molecules. Starch, glycogen, cellulose, and chitin are sometimes also referred to as glycans, but we will focus on smaller oligosaccharides that usually have more complex structures. Glycosylation, the attachment of glycans to proteins, is presumably one of the most extensive and complex protein PTM.

Monosaccharides (simple sugars) are the building blocks of glycans. Table 14.1 lists monosaccharides commonly found in higher animals; others can be found in bacteria and plants. A large number of monosaccharides exist, but only few are present for an individual species or cell: For example, humans express only NeuAc but not NeuGc, because of a missing enzyme [42]. Molecular formulas and masses in Table 14.1 are reported for monosaccharide residues; for the corresponding monosaccharide, add H_2O or 18.010565 Da. We see that monosaccharides

Figure 14.1: Open and cyclic form of the monosaccharide glucose: The chain form of D-glucose (left) and α-D-glucopyranose (right). **[TODO: REDRAW GRAPHICS, KEINE FARBEN, DAS KOMISCHE α WEG.]**

are mostly made up from carbon, hydrogen, and oxygen. Note the large number of isomers: Pentoses have five carbon atoms in the backbone, and all share the molecular formula $C_5H_{10}O_5$. Similarly, hexoses with six carbon atoms in the backbone have molecular formula $C_6H_{12}O_6$.

The following paragraph explains how glycans are formed from monosaccharides. As our algorithms deliberately ignore linkage types and only focus on glycan topology, one does not have to understand this paragraph in order to understand the algorithms in this chapter. For us, the important fact is that each monosaccharide has one "in-link" (the anomeric carbon) and usually four "out-links" (carbon hydroxy groups), and that monosaccharides can be glued together via these links (glycosidic bonds). But for the sake of completeness, let us have a look at how glycans are formed from monosaccharides. We will ignore chirality in our presentation, as this does not modify the masses of monosaccharides. See Chapter 2 of [227] for details.

Free monosaccharides can exist in an open or cyclic form, see Fig. 14.1. Carbon atoms are consecutively numbered, starting with the aldehyde carbon atom C-1 double-bonded to an oxygen atom. In glycans, we only find the cyclic form of monosaccharides: Here, the monosaccharide has a ring structure with five or six covalent bonds, made from carbon atoms and one oxygen atom. See again Fig. 14.1 (right) for glucose: atoms C-1 to C-5 plus one oxygen make up the ring of the cyclic monosaccharide. Two monosaccharides can be concatenated by a glycosidic bond, that is formed between the anomeric carbon of one monosaccharide and a hydroxy group of another: Chemically speaking, the hemiacetal group of one monosaccharide reacts with the alcohol group of the other monosaccharide, releasing water. In Fig. 14.1 (right), C-1 is the anomeric carbon, and C-2, C-3, C-4, and C-6 have hydroxy groups. This results in different linkage types, denoted "1–2" for a glycosidic bond involving anomeric carbon C-1 in the first monosaccharide, and carbon C-2 with a hydroxy group in the second monosaccharide. In a glycan, the unique monosaccharide that is not engaged in a glycosidic bond via its anomeric carbon, may be attached to a protein (see below) or a lipid. This is the distinguished *reducing end* of the glycan: Precisely speaking, "the reducing end of the oligosaccharide bears a free anomeric center that is not engaged in a glycosidic bond and thus retains the chemical reactivity of the aldehyde" [227]. It is still being referred to as reducing end, if the monosaccharide is in fact linked to, say, a serine or threonine.

The following is a rough classification of glycans:

N-glycans (or N-linked glycans) are attached to an asparagine residue of a protein or peptide. The amino acid sequence an N-glycan can be attached to, is either asparagine-X-serine or asparagine-X-threonine, where X is any amino acid except proline. All N-glycans

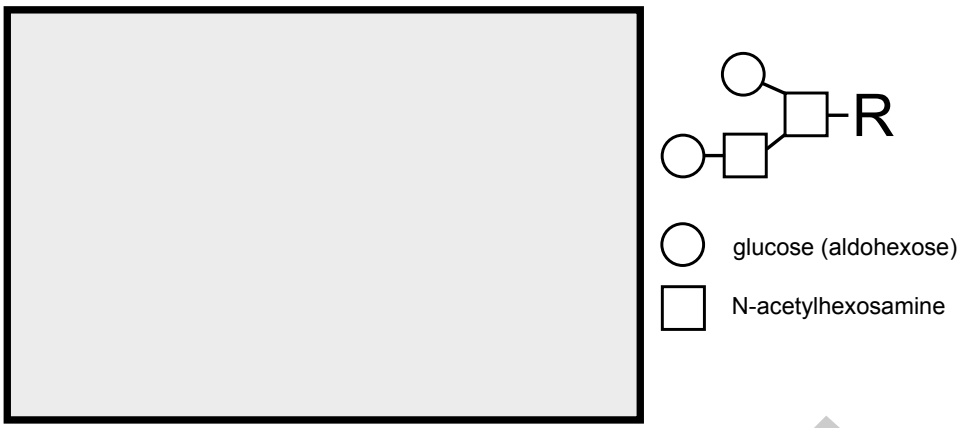

Figure 14.2: Structural formula (left) and topology (right) of a glycan made from four monosaccharides. **[ToDo: Draw figure!]**

are derived from a common precursor, which is then extensively modified. Still, unlike O-glycans, all N-glycans share a rather similar topology. N-glycans are important for protein folding, among others, and they are very common in eukaryotes but less common in prokaryotes.

O-glycans (or O-linked glycans) are attached to a serine or threonine residue of a protein or peptide. O-glycan assembly starts with an N-acetyl-galactosamine monosaccharide. Unlike N-glycans, there is no common precursor, and at least four "core structures" are known. O-glycans are very common in eukaryotes but less common in prokaryotes.

Glycosaminoglycans have a linear topology and contain long repetitions of disaccharide motifs. They are attached to a protein or peptide via an O-link.

Free glycans are not attached to anything. They are used as signaling molecules for a variety of biological processes, such as plant defense response. Also, free glycans are found in the milk of mammals, and glycans found in human milk appear to protect infants against pathogens affecting the intestines.

So much for the biochemistry; let us come back to computational mass spectrometry, introducing a formal model for glycan topologies. Unlike for peptide sequencing, the alphabet of monosaccharides (the glycan building blocks) can differ depending on the type of glycan we are analyzing. We assume that the alphabet Σ of monosaccharides is fixed and provided by the user, based on the biological background of the experiment. Every element $g \in \Sigma$ is assigned a residue mass $\mu(g)$. At this level, no monosaccharide isomers can be differentiated; depending on our background knowledge about the glycan, we may assume either hexose (Hex) or, say, glucose (Glc) to be part of our alphabet Σ.

We model a *glycan topology* as a rooted tree $T = (V, E)$. The root of the tree is the distinguished root monosaccharide, which can be attached to a protein or peptide. Tree vertices are labeled with monosaccharides from Σ and, hence, each vertex in the tree is also assigned a mass. Every vertex has an out-degree of at most four, because each monosaccharide has at most five linkages. See Fig. 14.2 for an example. We will use the letter T to denote both the underlying tree structure, and the glycan topology that includes vertex labels. To find the molecular formula or

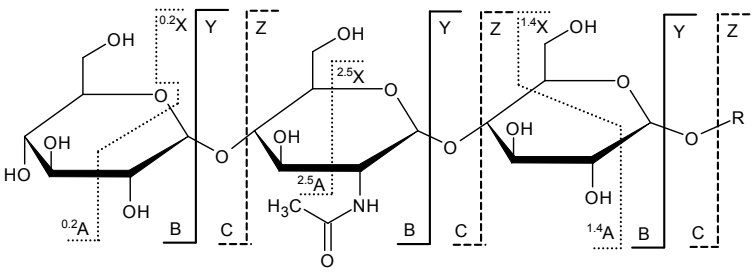

Figure 14.3: Fragments resulting from tandem mass spectrometry analysis of a glycan. Note that B, C, Y, and Z ions are not affected by linkage types. **[TODO: REDRAW FIGURE!]**

mass of a glycan topology we simply add up the molecular formulas or masses of the constituting monosaccharides and, finally, add H_2O or 18.010565 Da. Note that glycan topologies do not contain information regarding linkage types.

Our method will take into account all possible glycan topologies, deliberately ignoring all biological restrictions on, say, the amount of branching in the tree. It is well known that certain branching types are observed seldom in biological samples: For example, most monosaccharides show only one to three linkages, so most vertices in a glycan tree will have out-degree of at most two. But instead of completely forbidding such structures, we can incorporate biological restrictions into our scoring model, by subtracting a penalty if a structural rule is violated. In this way, we do not impede the discovery of rare structures that may diverge significantly from structural restrictions.

14.2 Glycan fragmentation

There are three types of fragmentation that break the glycan topology, resulting in six types of ions, see Fig. 14.3: X, Y, and Z ions correspond to fragments that contain the reducing end of the glycan and are called *reducing end ions* or *reducing end fragments*. A, B, C-ions, in contrast, do not contain this monosaccharide. A and X-ions are cross-ring fragments that result from internal monosaccharide breakages; the exact breakage positions are denoted by an additional superscript.

In the following, we assume that we have recorded a fragmentation spectrum of a single glycan. For glycans, collision-induced dissociation (CID) is often used as fragmentation technique. The collision energy determines the collision strength: The higher the energy, the more and stronger atomic bonds break. Since glycosidic bonds between sugars are weak compared to bonds inside the monosaccharides, we can choose the energy so that mainly these bonds break. This will predominantly generate B and Y ions, and we concentrate on these two types in our presentation.

We have modeled a glycan topology as a rooted tree $T = (V, E)$, where vertices are labeled with monosaccharides from an alphabet Σ. A fragment T' of T is a connected subtree, and the mass of T' is the sum of masses of the constituting vertices. Let $M := \mu(T)$ be the *parent mass* of the glycan structure. To simplify our presentations, we ignore mass modifications, such as adding the terminal H_2O group, reducing end modifications, the proton mass, or multiple ion series. As for peptides, these modifications can be easily incorporated into our method, see Sec. 14.5 below.

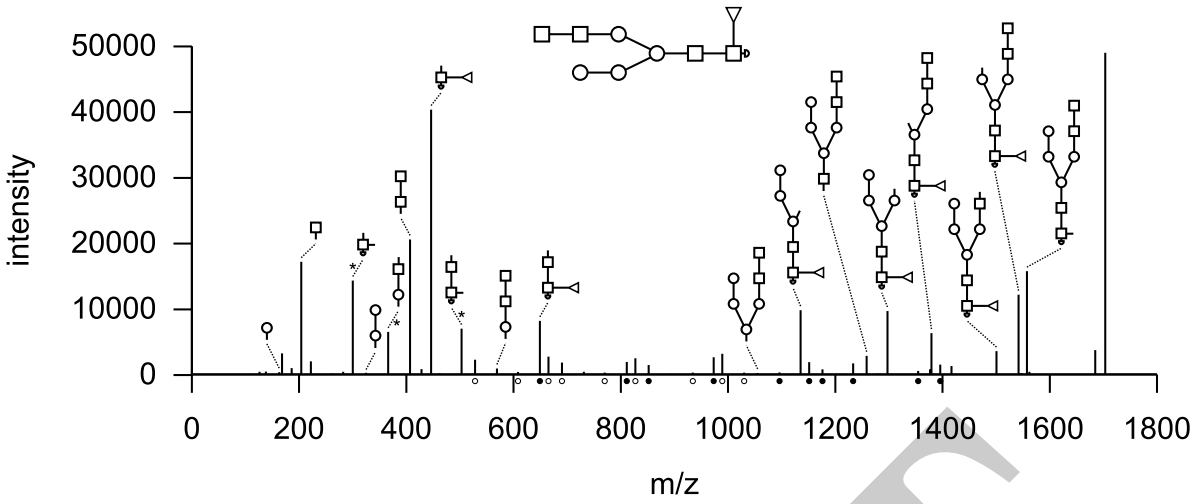

Figure 14.4: The peak-picked spectrum of a glycan with four hexoses, four N-acetylhexosamines, and one fucose. The spectrum is annotated with single-cleaved fragments of the correct glycan topology, and all but one peaks corresponding to single-cleaved fragments are found in the spectrum. Three intense peaks marked with (*) correspond to double-cleaved fragments. Additionally, all peaks marked with filled circles can be annotated with double-cleaved fragments of the glycan, and peaks marked with unfilled circles can be annotated with fragments that stem from more than two cleavages. Figure taken from [28]. **[TODO: ERLAUBNIS EINHOLEN?]**

Note that depending on the experimental setup and the glycans we are looking at, reducing end fragments may carry a peptide or peptide part as their "mass modification". This will make it easier for us to differentiate between the B and Y ion series.

For candidate generation, we restrict ourselves to simple fragmentation events, where only a single glycosidic bond is broken. This is a realistic assumption in application, see Fig. 14.4: By choosing an appropriate fragmentation energy, we can ensure that intense peaks usually correspond to single-cleaved fragments. Formally, such fragmentation is equivalent to removing a single edge. Hence, we can represent each simple fragmentation event by a vertex $v \in V$, where the subtree $T(v)$ induced by v represents the non-reducing end fragment, and the remainder of the tree is the reducing end fragment. The resulting non-reducing end fragments have the mass of a subtree of T induced by a vertex v, denoted $\mu(v)$. For reducing end fragments we subtract $\mu(v)$ from the parent mass M.

14.3 The candidate generation problem

In this section, we formalize the problem of glycan candidate generation: given the experimental data, we want to generate a small set of candidate glycan topologies, containing the correct topology. As in Chapter 2, we will use the peak counting score to compare hypothetical spectra with the measured one. A more involved scoring using, say, peak intensities is again considered at a later stage, namely Sec. 14.5 below. Also, we will concentrate on finding the best topology, and generating sub-optimal topologies (our candidates) comes "for free", as we will be using dynamic programming once more.

To simplify our presentation, let us assume for the moment that all our mass spectra consist of non-reducing end ions only. It turns out that we can easily generalize our solution to include reducing end ions. Also, we assume that all fragments stem from single fragmentation events. For the moment, we assume all masses to be integer.

Assume we are given a glycan topology T, and we want to evaluate T against the measured spectrum. We use a simple fragmentation model to generate a hypothetical *candidate spectrum*, and count the number of shared peaks between the measured spectrum and the candidate spectrum. Let $f(m)$ be the characteristic function of the measured spectrum, telling us if a peak is present (then, $f(m) = 1$) or absent (then, $f(m) = 0$) in the measured spectrum at mass m. Summing $f(m)$ over all peak masses m that are present in the candidate spectrum, we count all peaks that are common to both the measured spectrum and the candidate spectrum. Clearly, we can replace this peak counting score by something more elaborate at a later stage. Formally, we define

$$S(T) := \sum_{m=0,\ldots,M} f(m) \cdot g_T(m) \tag{14.1}$$

where

$$g_T(m) := \begin{cases} 1 & \text{if } T \text{ contains some subtrees } T(v) \text{ with mass } \mu(v) = m \\ 0 & \text{otherwise} \end{cases} \tag{14.2}$$

is the characteristic function of the glycan topology T, telling us if the tree contains a subtree of a certain mass. The important point is that $g_T(m) = 1$ holds if there is *at least* one subtree of mass m, independent of the *actual number* of such subtrees. Now, the GLYCAN CANDIDATE GENERATION problem can be stated as such: Find a glycan topologies T^* such that $S(T^*)$ is maximum; and afterwards, find all glycan topologies T such that

$$S(T) \geq (1-\varepsilon)S(T^*)$$

for some fixed $\varepsilon > 0$. This is the set of glycan topology candidates that is passed to the evaluation step of our sequencing algorithm.

Unfortunately, it turns out that finding an optimal topology T^* is an NP-hard problem: Precisely speaking, the decision problem "is there a glycan topology T such that $S(T) \geq t$ for some threshold t?" is NP-complete, even if we restrict ourselves to binary trees, where each monosaccharide vertex has at most two children. So, there is little hope for an algorithm with running time polynomial in M; unless P = NP, no such algorithm can exist. This makes glycan sequencing quite different from peptide sequencing, where this oversimplified version of the problem is comparatively easy to answer, see Chapter 2.

Let $T = (V, E)$ be a glycan topology. We introduce another scoring model, namely

$$S'(T) := \sum_{v \in V} f(\mu(v)) \tag{14.3}$$

which compares the measured spectrum, encoded by the characteristic function f, with a candidate glycan topology. Unfortunately, S' is not a peak counting score. Instead, for *every* subtree T' of T with mass $m' = \mu(T')$ we add $f(m')$ to the score. In this way, a glycan topology that contains many subtrees of identical mass m' receives a high score if $f(m')$ is large, even if it ignores all other peaks; see for example Exercise 14.5. We will show in the next section how computations for this model can be transferred over to peak counting scores, though.

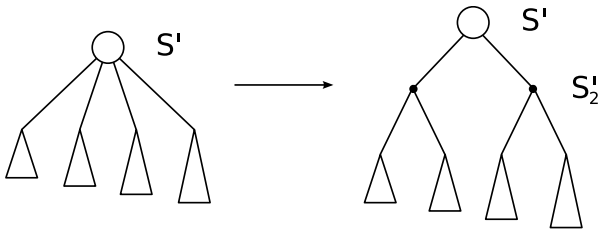

Figure 14.5: In (14.4) we compute the score for appending up to four previously computed subtrees to one monosaccharide, the root of the current subtree. Equation (14.5) reduces the complexity of computation by appending two "headless" subtrees to the root monosaccharide; each "headless" subtree, in turn, consist of two subtrees. **[TODO: REDRAW FIGURE!]**

To find the glycan topology T that maximizes the score $S'(T)$, we define $S'[m]$ to be the maximum score of any glycan topology with total mass m. We assume $S'[m] = -\infty$ if there is no glycan topology of that mass. It is easy to see that S' can be computed by the recurrence

$$S'[m] = f(m) + \max_{m_1+m_2+m_3+m_4+\mu(g)=m} S'[m_1]+S'[m_2]+S'[m_3]+S'[m_4], \tag{14.4}$$

where the maximum is taken over all $g \in \Sigma$ and $0 \le m_1 \le m_2 \le m_3 \le m_4 < m$; see Exercise 14.1. The term $S'[m]$ corresponds to any subtree of mass m with an arbitrary monosaccharide at its root. We initialize $S'[0] = 0$, as the "empty glycan topology" does not explain any peaks. We further assume $S'[m] = -\infty$ for all $m < 0$. But what about monosaccharide vertices that do not have the maximum out degree of four? Actually, these are already covered in (14.4): If one or more of the m_j in (14.4) equals zero, then the monosaccharide at the root of the subtree has less than four bonds. The maximum score of any glycan topology of parent mass M is $S'[M]$.

Unfortunately, computation of $S'[M]$ takes much too long using (14.4), see Exercise 14.2. Luckily, we can speed up computations considerably:

$$\begin{aligned} S'[m] &= f(m) + \max_{g\in\Sigma} \max_{m_1=0,\dots,\left\lfloor\frac{m-\mu(g)}{2}\right\rfloor} S'_2[m_1]+S'_2\left[m-\mu(g)-m_1\right] \\ S'_2[m] &= \max_{m_1=0,\dots,\lfloor\frac{m}{2}\rfloor} S'[m_1]+S'[m-m_1] \end{aligned} \tag{14.5}$$

The term $S'_2[m]$ corresponds to a "headless" subtree without a monosaccharide at its root, see Fig. 14.5. See Exercise 14.3 regarding the correctness of this recurrence. Now, instead of attaching four children to a monosaccharide vertex, we attach two headless subtrees with two children each. Remember that the special case of less than four children is covered in (14.4), and it is also covered here.

Using (14.5) we can compute $S'[M]$ in time $O(|\Sigma| \cdot M^2)$: We have to compute M entries in S' and S'_2; computing $S'[m]$ requires $O(|\Sigma| M)$ time, and computing $S'_2[m]$ requires only $O(M)$ time. The actual algorithm is simply a For-loop over all masses $m = 0,\dots,M$, we omit the simple details. It should also be understood how to recover an optimal solution using backtracing, see Exercise 14.4. We reach:

Lemma 14.1. *Given a monosaccharide alphabet Σ with masses $\mu : \Sigma \to \mathbb{N}$, a parent mass M, and a function $f : \{0,\dots,M\} \to \mathbb{R}$ encoding the measured spectrum. Then, we can compute $S'[m]$*

and $S'_2[m]$ *for all* $m = 0,\ldots,M$ *in time* $O(|\Sigma| \cdot M^2)$ *using recurrence* (14.5)*. Next, we can recover a glycan topology* $T = (V,E)$ *maximizing* $S'(T)$ *in* $O(|V| \cdot M)$ *time, backtracing trough* S' *and* S'_2*.*

14.4 An exact algorithm for glycan candidate generation

So, the problem of generating glycan topologies from tandem MS data is NP-hard, even if we restrict ourselves to the simple peak counting model — what can we do? For obvious reasons, we want to stick with the dynamic programming approach, as this allows us to generate an arbitrary number of suboptimal solutions. Also for obvious reasons, we do not want to rely on heuristics at this early stage of our algorithm: If the true solution is missed during candidate generation, it cannot be brought back during candidate evaluation.

We now modify recurrences (14.5) to find the glycan topology T that maximizes $S(T)$. We do not want to take a closer look at the proof of NP-hardness in [208]; but we can assure the reader that the complexity of the problem only holds for mass spectra that contain a "large" number of peaks. Measured spectra, in contrast, are relatively sparse and contain only tens of peaks that have significant intensity: The number of simple fragments of a given glycan topology corresponds to the number of vertices in a tree, and this is only linear in the number of constituting monosaccharides. Let k be the number of peaks in the measured spectrum: k is the parameter of our problem, and we limit the running time explosion to this parameter, while maintaining a polynomial running time with respect to M. Choosing k to be *equal* to the number of peaks, is solely done for the ease of presentation. In the next sections, we show that parameter k can be arbitrarily chosen in application, trading specificity for running time and memory consumption of the method: For low k, the method produces more candidates that have a high score because of scoring peaks multiple times. This is comparable to the previous chapter, where we abandoned optimality to save time and space. It turns out that a moderate k, such as $k = 10$, is appropriate in practice [28].

In order to avoid multiple peak counting, we incorporate the set of explained peaks into the dynamic programming. Let C^* be the set of peak masses in the measured spectrum, where $|C^*| = k$. For every mass $m \leq M$ and every subset $C \subseteq C^*$ we define $S[C,m]$ to be the maximum score of any glycan topology T with total mass $\mu(T) = m$ where only the peaks from C are used to compute this score. At the end of our computations, $S[C^*,M]$ holds the maximum score of any glycan topology where all peaks from C^* are taken into account for scoring. We initialize $S[C,0] = 0$ for all $C \subseteq C^*$.

We could come up with a recurrence similar to (14.4) for computing $S[C,m]$ but this would be much too slow in practice, see Exercise 14.6. Instead, we modify the faster recurrence from (14.5) for our purpose: We define $S_2[C,m]$ to be the score of a "headless" glycan topology with mass m using only peaks in C. Again, we initialize $S_2[C,0] = 0$ for all $C \subseteq C^*$.

Now, S_2 helps us to restrict the branching in the tree to bifurcations: We limit the recurrence of $S[C,m]$ to two "headless" subtrees with disjoint peak sets $C_1, C_2 \subseteq C$, where C_1 is the subset of peaks explained by the first subtree, and C_2 is the set of peaks explained by the second subtree.

```
1: procedure GlycanSequencing(mass M, set of peak colors C*)
2:     Initialize S[C,0] ← 0 and S_2[C,0] ← 0 for all C ⊆ C*
3:     for m = 1,...,M do
4:         for all subsets C ⊆ C* do
5:             Compute S[C,m] using (14.6)
6:         end for
7:         for all subsets C ⊆ C* do
8:             Compute S_2[C,m] using (14.6)
9:         end for
10:    end for
11: end procedure
```

Algorithm 14.1: Generating glycan candidates: Besides the parent mass M and the set of peak colors C^*, the weighted alphabet Σ with integer masses $\mu : \Sigma \to \mathbb{N}$ and the function $f : \{0,\dots,M\} \to \mathbb{R}$ are inputs of the method.

We require $C_1 \cap C_2 = \emptyset$ what guarantees that every peak is scored at most once. Additionally, we demand $C_1 \cup C_2 = C \setminus \{m\}$. We obtain the following recurrences:

$$\begin{aligned} S[C,m] &= \max_{g\in\Sigma} \max_{m_1=0,\dots,\left\lfloor\frac{m-\mu(g)}{2}\right\rfloor} \max_{C_1\subseteq C\setminus\{m\}} \Big\{ f(C,m) + S_2[C_1,m_1] \\ &\qquad + S_2\big[C\setminus(C_1\cup\{m\}), m-\mu(g)-m_1\big] \Big\} \qquad (14.6) \\ S_2[C,m] &= \max_{m_1=0,\dots,\lfloor\frac{m}{2}\rfloor} \max_{C_1\subseteq C} S[C_1,m_1] + S\big[C\setminus C_1, m-m_1\big] \end{aligned}$$

What is $f(C,m)$? We have to delay the scoring of a peak at mass m if m is not in C: To this end, we define

$$f(C,m) := \begin{cases} 0 & \text{if } m \notin C \text{ but } m \in C^*, \\ f(m) & \text{otherwise.} \end{cases} \qquad (14.7)$$

So, both peaks not in C^* and peaks in C are scored, whereas scoring of peaks in $C^* \setminus C$ is delayed. For the particular case that C^* equals the set of all peaks (or peak colors) then the condition "$m \in C^*$" is always fulfilled. But as we have indicated above, we want this recurrence to work for an *arbitrary* set of peak colors.

We do not want to prove the correctness of recurrence (14.6), as this proof will be somewhat lengthy and technical. We just note that the presumably best way to prove the correctness, is to show the equivalence of (14.6) to a recurrence for four subtrees similar to (14.4), and then to show that this recurrence does indeed compute the correct values $S[C,m]$. See Exercise 14.7.

It is not very involved to compute (14.6) in an admissible order, see Algorithm 14.1. To see this, take a closer look at (14.6): Computing $S[C,m]$ accesses table entries $S_2[\cdot,m_1]$ with $m_1 < m$ but never for $m_1 \geq m$; furthermore, it does not access any entries $S[\cdot,\cdot]$. So, we ensure that $S_2[C,m_1]$ is has been computed for all $C \subseteq C^*$ and all $m_1 < m$, what is trivial, see the outer loop of Algorithm 14.1. Similarly, computing $S_2[C,m]$ will only access entries $S[\cdot,m_1]$ for $m_1 \leq m$, but not $S_2[\cdot,\cdot]$. So, we ensure that $S[C,m_1]$ is has been computed for all $C \subseteq C^*$ and all $m_1 \leq m$, what is again trivial: We simply interleave the computation of $S[\cdot,\cdot]$ and $S_2[\cdot,\cdot]$. See the previous section on how to implement an iteration over all $C \subseteq \{1,\dots,k\}$ and all $C' \subseteq C$. **[ToDo: Pass op!]**

We now analyze time and space that is needed for the computation of (14.6). One can easily see that the space required to store $S[C,m]$ is $O(2^k \cdot M)$. What is the time required to compute

all $S[C,m]$? It turns out to be $O(3^k \cdot |\Sigma| \cdot M^2)$. To understand this, we first note that factors $|\Sigma|$ and M come from the calculation of $S[C,m]$ in (14.6), and that we can ignore computation of $S_2[C,m]$ in our considerations. As we have to compute $S[C,m]$ for all $m = 0,\ldots,M$, another factor M is obvious. But we have to be careful about partitioning the set C: There are 2^k sets $C \subseteq \{1,\ldots,k\}$, and iterating over all $C_1 \subseteq C$ takes $O(2^k)$ time, resulting in a 4^k factor in our running time analysis. But this analysis is simply to imprecise: There exist 3^k possibilities to partition k peaks into the three sets C_1, C_2, and $C^* \setminus (C_1 \cup C_2)$. Every such partition will be evaluated exactly *once* in the computation of $S[C_1 \cup C_2, m]$, so *any* algorithm that computes (14.6) in a reasonable fashion, and in particular Algorithm 14.1, has the desired running time of $O(3^k \cdot |\Sigma| \cdot M^2)$.

To recover an optimal solution, we backtrace through the dynamic programming matrix starting from entry $S[C^*,M]$. It is obvious that the maximum score will explain as many peaks as possible, but not necessarily all. We can also compute sub-optimal solutions that deviate at most $\delta := \varepsilon \cdot S(T^*)$ from the optimal score $S(T^*)$ for some $\varepsilon > 0$, see Sec. 8.4. This backtracking usually generates many isomorphic trees, which have to we remove from the final output: This can be achieved by encoding glycan topologies as strings, see Exercise 14.8. Running time of backtracking is $O(out \cdot 2^k \cdot Mn)$ where n is the maximum size of a glycan topology in the output, and *out* is the number of generated trees including isomorphic trees, that is usually larger than the size of the final candidate set.

14.5 Mostly old wine in new skins

In this section, we present several modifications to the algorithm from the previous section, most of which are already known to us from the problem of peptide sequencing. Hence, the title. We will reference to ideas and tricks throughout this book, which might be slightly uncomfortable for the reader, but illustrates the degree to which we can "recycle" these ideas.

First, we can correct the parent mass, as described in Sec. 8.1. Second, it is understood that we can use a more elaborate scoring then the peak counting score; in particular, we can use peak intensities and peak masses as described in Sec. 4.2, see also Sec. 8.5. Third, we can use real-valued masses for the scoring, as described in Sec 8.5.

Next, we get rid of the unrealistic assumption that only non-reducing end ions (B ions) are present in the mass spectra. But whenever we find a non-reducing end ion then, for ideal data, the complementing reducing end ion must also be present in the mass spectrum for perfect data. We can "mirror" the spectrum in a preprocessing step, see Sec. 2.5.3. The spectrum now contains reducing end and non-reducing end peaks with same intensity for every observed peak, even if only one was detected by the instrument. But how do we avoid multiple peak counting for intense peaks? Some glycan topology might contain both an B and a Y ion of identical mass, and if a peak with this mass is present in the measured spectrum, we must not score it twice. In fact, avoiding multiple peak counting comes "for free": We regard the elements in C^* as colors, and assign complementing reducing end and non-reducing end ion peaks the same color. This can be achieved without changing recurrence (14.6) or Algorithm 14.1. This ensures that each peak is only scored once, either as a reducing end ion, or as a non-reducing end ions, whatever leads to the better score. In practice, this problem is usually not as daunting as for peptide *de novo* sequencing: the mass of a reducing end ion is often not decomposable if regarded as non-reducing end ion mass (and vice versa) because of reducing end modification, and will not be considered for the score anyway. This can be contributed to the small monosaccharide alphabet,

where many masses cannot be decomposed, see also below. Recall that in some case, the mass modification of the reducing end can be a peptide or peptide fragment. Here, the complementing ion series, such as B and Y ions, will show only small overlap, further simplifying the problem.

Similar to Sec. 8.6 we can also incorporate C and Z ions into our score, merging their intensities with those of the corresponding B and Y ion pair. By this, we are loosing our "peak counting score" interpretation of the optimization, but this should not be regarded a problem in applications.

We do not penalize for additional peaks that are not explained by our glycan, see Section 4.3 for a justification. Recall that for candidate generation, this is a sensible thing to do, as our scoring does not take into account peaks from multiple-cleaved fragments, or the A/X ion series. For our computations, this implies that we can safely ignore additional peaks, and we have to compute $S[C,m]$ and $S_2[C,m]$ only for those sets $C \subseteq C^*$ that do not contain any additional peaks. This can be efficiently implemented by using hash maps to store these values, and by restricting (14.6) to initialized entries, compare to Section 17.3. The maximum score is no longer found in entry $S[C^*,M]$ which is now usually undefined; instead, we search for the maximum entry $S[C,M]$ with $C \subseteq C^*$, and we have to start backtracing from this entry. Finally, when computing (14.6), sets C containing masses bigger than the current mass m need not be considered.

In case the number of peaks in a glycan mass spectrum is too large, we can easily limit the exponential growth in memory and running time by choosing an appropriate k such as $k = 10$. Now, we use the k most intense peaks C^* in our explanation at most once, whereas we allow all other peaks to contribute multiple times to the score. Compare to Sec. 13.5.

Using these engineering techniques, the prohibitive factor in the running time may easily become M^2. But we noted above that many masses m are not decomposable at all over the alphabet of monosaccharide masses, as our alphabet Σ is usually small in applications. Using sparse dynamic programming, we can exclude these masses from our computation, since there exists no subtree which could explain them. A similar argumentation shows that the mass remainder $M - m$ must also be decomposable over the alphabet of monosaccharide masses, so we can also exclude many masses close to M from our computations. Doing so, we again reduce running time and memory requirements of the algorithm in practice.

We can also include a-priori probabilities for the number of branches a monosaccharide has. For example, fucose generally does not connect to further monosaccharides. These probabilities can be estimated from known glycan structures. All recurrences presented in this chapter and, in particular, (14.5) and (14.6) can be modified to take into account properties of the monosaccharide g, such as the number of links of g for the scoring: This is obvious for the "long" recurrence (14.4) but can also be achieved for the recurrences using "headless" subtrees, see Exercise 14.9. Be aware that care has to be taken when learning branching probabilities: a particular branching may be rare in general but common at a certain position of the glycan, and the scoring must not impede such branching. As an example, we mention mannose in the core structure of certain N-glycans, with out-degree three.

14.6 Evaluation of glycan topology candidates

[TODO: MUCH SHORTER, SAY MOSTLY WHAT GOLDBERG *et al.* [93] HAVE DONE — OUR SCORING IS SOMEWHAT PRELIMINARY!]

Once we have reduced the set of potential glycan topologies from the exponential number of initial candidates, to a manageable set of tens or hundreds of structures, we can now evaluate each candidate glycan topology using an in-depth comparison between its theoretical spectrum and the measured spectrum. This comparison can also take into account multiple-cleaved fragment trees, C and Z ions, or A and X ions where a monosaccharide is cleaved. Since evaluation of candidate glycan structures is not the focus of this work, we only present a rather general scoring scheme, and we will not go into too much detail here. Still, we reach good identification results. Our scoring generalizes ideas of Goldberg *et al*. [93]. We stress that our evaluation approach leaves room for improvement.

The idea of our approach is to determine and score the fragmentation tree of the glycan, a representation of the consecutive fragmentation events. Therefore, we construct a fragmentation graph similar to the one used for metabolite identification in [26]. The fragmentation graph enables us to score all peaks that we can explain by fragmentation of the candidate glycan. Additionally, we can incorporate relationships between fragments. For example, a double-cleaved fragment receives a higher score if the intermediate single-cleaved fragment exists. We again avoid peak double counts by regarding peaks as colors, and not allowing to score a color twice. The *fragmentation graph* is constructed as follows: It contains a vertex for every subtree of the candidate tree whose mass deviates less than the instruments mass accuracy from a peak mass. Vertices whose subtrees correspond to the same peak, are colored with the same color. Vertices are connected by a directed edge if the tree of the descendant vertex is a subtree of the tree of the ancestor vertex. This results in a transitive colored digraph.

We now compute scores for the vertices and edges of the fragmentation graph. In contrast to candidate generation, we use the unfolded spectrum as there are separate vertices for reducing end and non-reducing end ions in the fragmentation graph. The vertex score comprises peak intensity and mass deviation. Additionally, we penalize for the number of fragmentation events necessary to produce the fragment from the fragmentation graph root. We refer to this number as the *fragmentation distance* x_r, and score it by a slowly falling function we chose as $\text{frag}_r(x_r) = 0.75^{x_r}$.

The edge score takes into account the fragmentation distance to the *parent vertex*. We assume that intermediate fragments should be observed in the spectrum. So, edges that represent multiple fragmentation events shall contribute less to the score than edges representing a single fragmentation event. We achieve this by using the sub-additive function $\text{frag}_e(x_e) = 1/x_e^2$, where x_e is the fragmentation distance of parent and child vertex. This function has been chosen *ad hoc* to avoid over-fitting to the data. To simplify the score calculation for the algorithm we pass on the vertex scores to all incoming edges, multiplying it with the edge scores. The overall score of an edge is then $s(e) = int \cdot erfc \cdot \text{frag}_r(x_r) \cdot \text{frag}_e(x_e)$, where *int* is the peak intensity and *erfc* is the complementary error function of the mass deviation.

The *maximum colorful subtree* [26] of this weighted graph (the subtree that has maximum weight and uses at most one vertex per color) then is a hypothetical fragmentation tree. Unfortunately, finding this tree is NP-hard, and we refrain from using the exact FPT algorithm from [26] since it is too slow to be executed for each candidate. Therefore, we apply a greedy heuristic to determine a score: This heuristic tests all edges in descending order of score. If an edge can be added violating neither the tree nor the colorful property, it is attached to the partial tree. Due to the transitivity of the fragmentation graph, this procedure eventually results in a tree. The score of the candidate structure is the sum of the edge scores of the fragmentation tree.

We incorporate C/Z ions by creating vertices not only for every B and Y ion subtree mass that could explain a peak, but also for those masses shifted by the mass increase or decrease of C or Z ions. These vertices are also colored according to the mass of the peak they explain. Edge creation is then performed as usual. This ensures that C and Z ions are considered separately from their corresponding B and Y ions, but a peak may only be explained by either a B/Y or a C/Z ion.

Glycans are composed of fucoses (F, mass 146.06 Da), hexoses (H, 162.05 Da), and N-acetylhexosamines (N, 203.08 Da). Average running time for candidate generation was 2.5 s without and 4.0 s including traceback. There were many spectra with less than 10 decomposable peaks what reduced running time. For the "loaded" spectra with $k = 10$, average running time was 3.6 s without and 5.5 s with traceback. In all except two cases the candidate set contained the manually determined topology, see below. For 17 spectra the true topology was found in the TOP 50 of candidates, and for 12 spectra even in the TOP 25 of candidates.

We also tested if avoiding peak double-counting is needed for candidate generation. To this end, we set $k = 0$, so every peak could be counted an arbitrary number of times. Doing so, candidate generation produced the correct topology only for eight of the 24 spectra even if the candidate set was chosen to contain at least 500 structures. This shows that avoiding multiple peak counting is essential for the analysis. Certain glycan topologies do in fact create the same fragment mass several times: It must be understood that our approach does not *penalize* such topologies, but it also does not *reward* them. As the extreme case, consider a single leaf of the tree: If the corresponding peak has high intensity, then we reward trees for having identical labels at all leaves, which is surely not desirable. Finally, we tested if further increasing k could improve the results of candidate generation. But as it turned out, increasing k to the 15 most intense peaks did not improve the results. So, computations can be carried out with a moderate k such as $k = 10$ for glycans of this size, without loosing specificity.

Once we have reduced the set of potential glycan topologies from the exponential number of initial candidates, to a manageable set of tens or hundreds of structures, we can now evaluate each candidate glycan topology using an in-depth comparison between its theoretical spectrum and the measured spectrum. This comparison can also take into account peculiarities of the mass spectrometry analysis, such as multiple-cleaved fragment trees, other ion series such as A/X and C/Z ions, or those X-ions that have lost parts of a monosaccharide.

14.7 Counting glycan topologies

We have mentioned above that the number of glycan topologies easily becomes prohibitive for enumerating all possible topologies. We now substantiate this claim, by computing the number of glycan topologies for a certain number of monosaccharides, and for a certain mass. We first recapitulate some classical results for counting rooted trees of bounded degree, then present an algorithm for the exact and swift counting of glycan topologies. Our analysis will be strictly combinatorial, as we will ignore whether geometrical constraints render certain glycan topologies impossible: These glycan topologies will show large amount of branching, particularly close to the root. In this sense, the number computed here, are upper bounds to the "true" number of glycan topologies; but these bounds are "sharp" in the sense that, from a combinatorial standpoint, our calculations will be exact. Taking into account considerations from molecular geometry will be extremely difficult: This might boil down to enumerating all

	approx.	number of glycan topologies				
n	using (14.8)	$\|\Sigma\| = 1$	$\|\Sigma\| = 2$	$\|\Sigma\| = 3$	$\|\Sigma\| = 4$	$\|\Sigma\| = 5$
1	1	1	2	3	4	5
2	1	1	4	9	16	25
3	2	2	14	45	104	200
4	4	4	52	246	752	1800
5	9	9	214	1485	5996	17850
6	19	19	904	9369	50288	186750
7	44	45	4038	61947	440784	2039500
8	105	106	18508	421668	3980384	$2.30 \cdot 10^7$
9	257	260	87008	2939562	$3.68 \cdot 10^7$	$2.64 \cdot 10^8$
10	639	643	416388	$2.09 \cdot 10^7$	$3.46 \cdot 10^8$	$3.10 \cdot 10^9$
15	72664	72917	$1.25 \cdot 10^9$	$4.51 \cdot 10^{11}$	$3.07 \cdot 10^{13}$	$8.24 \cdot 10^{14}$
20	9866231	9881527	$4.51 \cdot 10^{12}$	$1.16 \cdot 10^{16}$	$3.23 \cdot 10^{18}$	$2.61 \cdot 10^{20}$
25	$1.48 \cdot 10^9$	$1.48 \cdot 10^9$	$1.78 \cdot 10^{16}$	$3.29 \cdot 10^{20}$	$3.75 \cdot 10^{23}$	$9.08 \cdot 10^{25}$
30	$2.35 \cdot 10^{11}$	$2.35 \cdot 10^{11}$	$7.50 \cdot 10^{19}$	$9.89 \cdot 10^{24}$	$4.62 \cdot 10^{28}$	$3.36 \cdot 10^{31}$

Table 14.2: Number of glycan topologies for n vertices and an alphabet size of one to five, plus rounded approximation (14.8) corresponding to $|\Sigma| = 1$.

glycan topologies using the methods presented below, then testing for each glycan topology if it can exist in 3D space.

On the other hand, our computations do not take into account linkage types or chirality. It should not be to hard, though, to modify the methods of this section for that purpose, see Exercise 14.13.

Let $N[n, |\Sigma|]$ be the number of different glycan topologies with n vertices, where vertices are labeled with elements from Σ. One can easily see that this number does not depend on the actual alphabet Σ but only on its cardinality.[1] Recall that a glycan topology corresponds to a rooted tree such that every vertex has out-degree at most four. In the following a *glycan tree* is a rooted trees with out-degree at most four. Note that counting glycan trees does not take into account that vertices are labeled, and corresponds to the the case $|\Sigma| = 1$. See Table 14.2 and Fig. 14.6 below, for the number of different glycan topologies for an alphabet size of one to five.

Similar to Sec. 3.7, one can approximate the number of glycan trees using a closed formula: This number asymptotically behaves like

$$N[n, 1] \sim \tilde{t}(n) := 0.462103 \cdot 2.911038^n \cdot n^{-3/2}. \tag{14.8}$$

This approximation is very accurate even for small n, see Table 14.2: For $n = 10$ we calculate $\tilde{t}(10) = 638.6$ whereas the true number is 643, so the relative error is well below one percent.

We can estimate the number of glycan topologies over an *arbitrary* alphabet Σ by $|\Sigma|^n \cdot \tilde{t}(n)$, since every vertex can be colored with an individual color. This overestimates the number of glycan topologies, as we do not take into account isomorphic trees; see Exercise 14.10. As an example, consider $|\Sigma| = 5$ and $n = 10$: We estimate this number to be $5^{10} \cdot \tilde{t}(10) = 6.24 \cdot 10^9$ whereas the true number is $3.10 \cdot 10^9$, so the true number is only *half* of what our rough estimate tells us. The relative error will become even larger as n increases, see Exercise 14.11.

[1] We will not use k to denote the size of the alphabet in this section, as we have "consumed" it for the parameter k in the previous sections.

The "classical way" of counting trees, is to use Pólya's enumeration theorem [188]. We omit the details, but you eventually come up with the recurrence

$$t(n) = \frac{1}{24}\Big(\sum_{i+j+k+l=n-1} t(i)t(j)t(k)t(l) + 6 \sum_{i+j+2k=n-1} t(i)t(j)t(k) + 3 \sum_{2i+2j=n-1} t(i)t(j) + 8 \sum_{i+3j=n-1} t(i)t(j) + 6 \sum_{4i=n-1} t(i) \Big) \tag{14.9}$$

where $t(n) := N[n,1]$ is the number of glycan trees, and we initialize $t(0) = 1$. Using some simple tricks, one can compute $t(n')$ for all $n' = 1,\dots,n$ in $O(n^3)$ time [28]. Besides our crude estimate above, there seems to be no possibility to generalize this results to trees where nodes are (non-uniquely) labeled with elements from a finite set.

We now want to find a method for the exact computation of $N[n] := N\big[n,|\Sigma|\big]$, for a fixed alphabet Σ of arbitrary size. The only reason to leave out the '$|\Sigma|$' is to make things more readable. To reach an efficient recurrence, we distinguish four cases, corresponding to the out-degree of the root vertex. To the root vertex, we attach a forest of one to four trees, that in total have one vertex less than the new glycan topology. In our recurrence, we will also have to upper bound the number of vertices in each individual tree. To this end, let $N_i[n,k]$ be the number of forests consisting of i non-empty glycan trees, such that the total number of vertices in the forest is n, and no tree in the forest has more than k vertices. Clearly, we can label the root with any element from Σ. Hence, the number of glycan topologies labeled with the alphabet Σ is

$$N[n+1] = |\Sigma| \cdot \big(N[n] + N_2[n,n] + N_3[n,n] + N_4[n,n]\big). \tag{14.10}$$

We initialize $N[0] = 1$ for the empty tree.

Obviously, the main difficulty is to compute the number of forests $N_i[n,k]$: Assume that $l \le k$ is the maximum size of any tree in such a forest, and let $j \le i$ be the number of trees of size l in this forest. We can choose $\binom{N[l]+j-1}{j}$ different trees of size l. Now, $i-j$ trees remain in the forest, and these can have at most $l-1$ vertices each, and must have $n-jl$ vertices in total. From the definition above, we know that there exist $N_{i-j}[n-jl, l-1]$ such forests. Thus, we reach the recurrence

$$N_i[n,k] = \sum_{j=1}^{i} \sum_{l=\lceil n/i \rceil}^{\min\{k,\lfloor n/j \rfloor\}} \binom{N[l]+j-1}{j} \cdot N_{i-j}\big[n-jl, l-1\big] \tag{14.11}$$

for the number of different forests with i glycan trees, maximum tree size k, and n vertices in total. We have to initialize $N_i[n,k]$ depending on the number of trees i: For $i = 0$, we set $N_0[0,k] := 1$, and $N_0[n,k] := 0$ for all $n \ge 1$. For $i = 1$, we set $N_1[0,k] = 0$ for all $k \ge 0$, $N_1[n,k] := N[n]$ for $n \le k$, and $N_1[n,k] := 0$ otherwise. For $i \ge 2$, we set $N_i[n,k] := 0$ in case $i > n$ or $k \cdot i < n$ or $k = 0$ holds. All other values can be computed from recurrences (14.10) and (14.11). So, we have reached:

Lemma 14.2. *Using recurrences* (14.10) *and* (14.11)*, the number of glycan topologies with n vertices over an alphabet Σ can be computed in time $O(n^3)$ and space $O(n^2)$.*

One might argue that the number of glycan topologies that we can find in biology, will be much smaller, as we rarely observe that any monosaccharide links to five other monosaccharides. To this end, we can easily modify our calculations to count glycan topologies with maximal out-degree three, see Exercise 14.12. We have plotted the number of glycan topologies for different monosaccharide alphabet sizes, both with out-degree four and three, in Fig. 14.6. Again, we can

Figure 14.6: Number of glycan topologies for an alphabet size of one to five, plus rounded approximation (14.8) corresponding to $|\Sigma| = 1$. Solid lines correspond to the usual out-degree four, whereas dashed lines correspond to the reduced out-degree of three. Notice the log-scale of the y-axis. **[TODO: DRAW FIGURE!]**

approximate the number of glycan trees with out-degree at most three and $|\Sigma| = 1$ using a closed formula:

$$\tilde{t_3}(n) := 0.5178760 \cdot 2.815460^n \cdot n^{-3/2}. \tag{14.12}$$

(By this nomenclature, $\tilde{t}(n)$ from (14.8) should rather be denoted $\tilde{t_4}(n)$.) Finally, if you feel that out-degree three might still be too high for biological relevant glycans, here is the approximation for the number of glycan trees with out-degree at most two and $|\Sigma| = 1$:

$$\tilde{t_2}(n) := 0.791603 \cdot 2.483254^n \cdot n^{-3/2}. \tag{14.13}$$

Finally, we show how to count the number of glycan topologies of a given mass M. One may be tempted to compute all possible monosaccharide compositions of mass M, then use the multinomial coefficient of the composition times $N[n]$ to compute the number of labeled glycan trees for each monosaccharide composition. Unfortunately, this again overestimates the true number of trees, as we have to take into account that siblings may induce isomorphic trees, in which case the true number of labellings is much smaller. In addition, this approach becomes prohibitive for large masses and large alphabets, as the number of decompositions explodes.

Here, we combine (14.4) with the above recurrences (14.10), (14.11), to reach a recurrence that is practically independent of $|\Sigma|$. We again assume integer masses $M = 0, 1, 2, \ldots$. Let Σ be the alphabet of monosaccharides. Let $\mathcal{N}[M]$ be the number of glycan topologies over Σ with mass M. Finally, let $\mathcal{N}_i[M, m]$ be the number of forests consisting of i non-empty glycan trees, such that the total mass of the trees in the forest is M, and no tree in the forest weights more than m. Similar to (14.10) we have

$$\mathcal{N}[M] = \sum_{g \in \Sigma} \Big(\mathcal{N}[M - \mu(g)] + \mathcal{N}_2[M - \mu(g), M] + \mathcal{N}_3[M - \mu(g), M] + \mathcal{N}_4[M - \mu(g), M] \Big). \tag{14.14}$$

We initialize $\mathcal{N}[0] := 1$. For brevity, we assume $\mathcal{N}[M] = 0$ and $\mathcal{N}_i[M, m] = 0$ for $M < 0$.

Again, the main difficulty is to compute the number of forests $\mathcal{N}_i[M, m]$, what can be achieved by a variation of (14.11):

$$\mathcal{N}_i[M, m] = \sum_{j=1}^{i} \sum_{m' = \lceil M/i \rceil}^{\min\{m, \lfloor M/j \rfloor\}} \binom{\mathcal{N}[m'] + j - 1}{j} \cdot \mathcal{N}_{i-j}[M - jm', m' - 1] \tag{14.15}$$

Similar to above, we initialize $\mathcal{N}_i[M, m]$: For $i = 0$, we set $\mathcal{N}_0[0, m] := 1$, and $\mathcal{N}_0[M, m] := 0$ for all $M \geq 1$. For $i = 1$, we set $\mathcal{N}_1[M, m] := \mathcal{N}[M]$ for $M \leq m$, and $\mathcal{N}_1[M, m] := 0$ otherwise. For $i \geq 2$, we set $\mathcal{N}_i[M, m] := 0$ in case $m = 0$ or $i > M$ or $m \cdot i < M$. We reach:

Lemma 14.3. *Using recurrences* (14.14) *and* (14.15)*, the number of glycan topologies with integer mass M over an alphabet Σ can be computed in time $O(|\Sigma| M + M^3)$ and space $O(M^2)$.*

A nice feature of the above recurrence, is that we can also use arrays $\mathcal{N}[\cdot]$ and $\mathcal{N}_i[\cdot,\cdot]$ to *enumerate* all glycan topologies of mass M, similar to what we did for strings in Sec. 6.2; see Exercise 14.14. If you want to enumerate glycan trees with fixed number of monosaccharides, though, then there are faster and simpler ways of enumerating these trees than to use the recurrences from this section [148].

14.8 Glycan mass spectrometry programs

Existing approaches for glycan *de novo* sequencing follow the usual two-step approach of first generating candidates, then scoring them [71, 89, 93, 208, 222]. Several approaches have been developed through the years for *de novo* sequencing of glycans, although glycan sequencing cannot compete with peptide sequencing in this respect:

STAT by Gaucher *et al*. [89] first decomposes the parent mass of the glycan molecule, as well as the masses of all other peaks found in the spectrum. The user has to manually decide on one of the decompositions. STAT then generates all glycan topologies for the chosen decomposition, and uses certain restriction to rule out some topologies.

StrOligo by Ethier *et al*. [71] focuses on N-glycans. It is based on biological knowledge about N-glycan structure. It enumerates all biologically plausible N-glycan topologies for the given parent mass, and evaluates each topology based on the measured spectrum.

OSCAR by Lapadula *et al*. [141] is special in that it does not use MS^2 but MS^n fragmentation data as input. It is a true *de novo* approach, as it does not require prior biological information, and appears to show good performance in practice. Its main disadvantage is that it requires MS^n data.

GLYCH by Tang *et al*. [222] does not only derive the topology of the glycan, but also deduces linkage types using cross-ring ion fragments. GLYCH uses dynamic programming to compute the optimal score of a structure, which is similar in spirit to S' from (14.3). The drawbacks of this approach has been described above: certain peaks might be score many times by the dynamic programming, leading to somewhat arbitrary optimal solutions. In fact, it appears that GLYCH prefers linear structures to branched ones. In a later publication, Sheng *et al*. [209] concentrate on solely inferring linkage types.

CartoonistTwo by Goldberg *et al*. [93] focuses on O-glycans — note that this is a much more difficult problem than sequencing N-glycans, as the search space is much larger. The program generates all possible topologies, using biological constraints for O-glycans, and scores them using an elaborate scheme.

GlycoMaster by [208] **[ToDo: Pass op!]**.

Entries are in chronological order. Regarding candidate generation, the above programs can be subdivided into three categories: Some approaches enumerate all possible glycan topologies of the given parent mass [71, 89, 93]. This is possible as in application, the alphabet of monosaccharides is usually very small (three to five monosaccharides) and, hence, the number of decompositions is also small — quite often, there is only one decomposition. If glycans become larger, this again results in a combinatorial explosion of tree structures, see Sec. 14.7. To cope with this problem, tools apply strict biological rules to cut down on the number of candidates.

GLYCH uses dynamic programming similar to Sec. 14.3 but simply ignore the problem of multiple peak counting [222]. Finally, Shan *et al*. [208] present a heuristic that avoids peak double counting. Regarding scoring the candidates, the by far most elaborate approach is due to Goldberg *et al*. [93], who use dependencies between the observed fragments to modify the score.

Besides the (*de novo*) interpretation of tandem MS data, other approaches have been developed for glycan MS analysis:

Glycomod by Cooper *et al*. [45] allows the user to search for N- and O-linked glycans, searching in the Swiss-Prot and TrEMBL databases. Glycomod includes some structural constraints, but no structural assignment is provided. **[TODO: SINGLE-STAGE MS]**

GlycosidIQ by Joshi *et al*. [122]

GlycoFragment and GlycoSearchMS by Lohmann and von der Lieth [150] ... database searching

SimGlycan is a commercial software, developed by PREMIER Biosoft in Paolo Alto, USA, to search a database for glycan tandem MS spectra.

[ToDo: Cartoonist?] by [92, 95] ... single-stage MS

xxx by Goldberg *et al*. [94] .. combining single-stage MS and tandem MS data

Glyco-Peakfinder by Maass *et al*. [158] is a web-service to annotate glycan mass spectra with *compositional* information. It allows for a subsequent database search to assign structures to the calculated compositions.

xxx by Wu *et al*. [239] ... glycosylations as a PTM

GlycoWorkbench by Ceroni *et al*. [35] assists an expert with the manual interpretation of glycan mass spectra. It automates various tedious steps of this interpretation, and provides an easy-to-use graphical user interface. GlycoWorkbench can evaluate a set of candidate structures, provided by the user, against a measured tandem mass spectrum.

14.9 Historical notes and further reading

See Raman *et al*. [190] for an introduction to the field of glycomics. The book "Essentials of Glycobiology" by Varki *et al*. [227] can be accessed freely over the Internet, see `http://www.ncbi.nlm.nih.gov/pubmed/20301239`.

Glycomics is, in many aspects, still in the developing stage [22]. This means that bioinformatics tools for this field still have to be developed — you might see this as a chance, as you do not have to fight with the old bulls. This is particularly so for the analysis of MS and tandem MS data. Citing Packer *et al*. [180] from 2008: "Key to the efforts in structural analysis are bioinformatics tools that can rapidly interpret MS data. These tools will be able to examine mass profiles for composition and MS fragmentation spectra to provide rudimentary or complete structures. Software that can perform these tasks will greatly advance the efforts in glycomics as it has done in proteomics."

The term "glycan sequencing" has been repeatedly used in the literature for the structural elucidation of glycans; other terms include "extracting sequence information" [89], "glycan

structure determination" [93], "glycan structure elucidation", and "glycan structural assignment" [71].

The presentation in this chapter largely follows the paper by Böcker, Kehr, and Rasche [28]. Shan, Ma, Zhang, and Lajoie [208] introduced the simple scoring model from (14.3) as well as the efficient recurrence (14.5), and established that generating glycan topology candidates while avoiding peak double counts is an NP-hard problem.

See Zaia [243] for a review on mass spectrometry analysis of glycans. The nomenclature of ion series is due to Domon and Costello [58].

Similar to the previous chapter, the running time of our dynamic programming algorithm in Sec. 14.3 can be reduced from $O(3^k \cdot |\Sigma| M^2)$ to $O(2^k \cdot k^2 |\Sigma| M^2)$ using **[ToDo: something]** [21]. Again, the practical use seems to be very limited due to the required overhead.

The number of glycan topologies with $|\Sigma| = 1$ or, equivalently, the number of rooted trees where each vertex has out-degree at most four, is sequence A036718 in the On-Line Encyclopedia of Integer Sequences.[2]

Approximations (14.8) and (14.12) are due to Otter [178], who also showed how to approximate these numbers for arbitrary fixed out-degree d. Unfortunately, he left out the constants for rooted trees with maximal out-degree four, see (14.8). Note that the multiplicative constant of (14.13) is correct [81, page 477], but different from the one in [178]. If you want to find approximations for pretty much any type of trees, take a look at Flajolet and Sedgewick [81] and Harary, Robinson, and Schwenk [107]. See also Exercise 14.15.

The findings of Sec. 14.7 can be generalized to counting arbitrary rooted trees with out-degree bounded by d [28]. Note that these results do not apply for glycans, but for other structures that can be viewed as rooted trees of bounded degree, for example, aliphatic amino acids [103].

Li and Ruskey [148] show how to enumerate trees with bounded degree, see Sec. 2.2 of their paper. The amortized running time per tree is "constant for realistic values of n", such as $n \le 25$, where n is the number of vertices in the tree.

14.10 Exercises

14.1 Proof that recurrence (14.4) is correct, that is, $S'[m]$ computed using this recurrence is truly the maximum score of any glycan topology with total mass m.

14.2 Establish the running time of computing $S'[M]$ using (14.4).

14.3 Show that computations using (14.4) and (14.5) will lead to identical results.

14.4 Show how to recover an optimal solution from the array S' using backtracing. What is the running time?

14.5 Assume that we have two monosaccharides $\Sigma = \{\mathsf{F}, \mathsf{H}\}$ with masses $\mu(\mathsf{F}) = 146$ and $\mu(\mathsf{H}) = 162$. Assume further that we have recorded the mass spectrum (without ion series modifications)

$$\mathcal{M} = \{146, 162, 454, 338, 938\}$$

with parent mass $M = 938$. Assume further that the peak at mass 146 has intensity 2, whereas all other peaks have intensity 1. Assume that we consider peak intensities in the

[2] `http://oeis.org/A036718`

scores $S(M), S'(M)$ as proposed in Sec. 8.5; so, the peak at mass 146 scores twice as much as the other peaks. What is the glycan topology maximizing $S(M)$; and what is the glycan topology maximizing $S'(M)$?

14.6 What is the running time of computing $S[C,m]$ by a recurrence similar to (14.4)? Note the base of the exponential growth!

14.7★ Proof the correctness of recurrence (14.6). You can do so by first showing its equivalence with a recurrence similar to (14.4), compare to Exercise 14.6; then, showing that this recurrence will compute the correct values.

14.8 Describe an algorithm to transform a glycan topology into string, that *uniquely* describes the glycan topology. The important point here is that for glycan topologies, the children of a monosaccharide are unordered, and we have to give them a "canonical ordering."

14.9 It is obvious that we can score the "degree of branching" in (14.5), by counting the number of m_i with $m_i = 0$. It is slightly less obvious how to achieve this for (14.5) and (14.6) while at the same time, guaranteeing the optimality of the computation. Discuss the problem for (14.5), and show how it can be modified accordingly. Hint: If $S'_2[m]$ is optimal for $m_1 = 0$, then $S'_2[m] = S'[m]$.

14.10 We noted in Sec. 14.7 that $N[n,|\Sigma|] \approx |\Sigma|^n \cdot t(n)$ overestimates the true number of trees. Show, for $n = 5$ and $\Sigma = \{\mathsf{F}, \mathsf{H}\}$, what glycan topologies are counted by this estimate, and which of them should not be counted (multiple times).

14.11 Calculate the exact numbers $N\big[n,|\Sigma|\big]$ for $n = 1,\ldots,30$ and $|\Sigma| = 1,2,3,4,5$. Compare the "rough estimate" $|\Sigma|^n \cdot \tilde{t}(n)$ to the exact number, and calculate the relative error.

14.12 Modify recurrence (14.10) and (14.11) so that the out-degree of each monosaccharide is at most three. Hint: The answer is mostly trivial.

14.13★ We also want to take into account different linkage types when counting glycan topologies: To this end, assume that $d\_(g)$ denotes the number of possible linkage points for monosaccharide $g \in \Sigma$. Modify recurrence (14.10) and (14.11) to take into account linkage types. Note that we no longer have to consider isomorphic subtrees in that calculation.

14.14 Use arrays $\mathcal{N}[\cdot]$ and $\mathcal{N}_i[\cdot,\cdot]$ from recurrence (14.14) and (14.15), to enumerate all glycan topologies of mass M.

14.15★ The constants in approximations (14.8), (14.12), and (14.13) only have six decimal places. Find more accurate constants, as described in point VII.21 on page 477–479 of Flajolet and Sedgewick [81]. Alternatively, you can use the 20-step recipe from Harary *et al*. [107].

Bibliography

[1] A. Aaant. I need a title, quick. **[TODO: REPLACE WITH A REAL CITATION]**, 2101.

[2] G. Alves, A. Y. Ogurtsov and Y.-K. Yu. RAId_DbS: peptide identification using database searches with realistic statistics. *Biol. Direct.*, 2:25, 2007.

[3] S. Andreotti, G. W. Klau and K. Reinert. Antilope – a lagrangian relaxation approach to the *de novo* peptide sequencing problem. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2011. To appear, doi:10.1109/TCBB.2011.59.

[4] R. Apweiler, H. Hermjakob and N. Sharon. On the frequency of protein glycosylation, as deduced from analysis of the SWISS-PROT database. *Biochim. Biophys. Acta*, 1473(1): 4–8, 1999.

[5] G. Audi, A. Wapstra and C. Thibault. The AME2003 atomic mass evaluation (ii): Tables, graphs, and references. *Nucl. Phys. A*, 729:129–336, 2003.

[6] J.-M. Autebert, J. Berstel and L. Boasson. Context-free languages and pushdown automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 111–174. Springer, 1997.

[7] V. Bafna and N. Edwards. SCOPE: A probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, 17:S13–S21, 2001.

[8] D. A. Barkauskas and D. M. Rocke. A general-purpose baseline estimation algorithm for spectroscopic data. *Anal. Chim. Acta*, 657(2):191–197, 2010.

[9] C. Bartels. Fast algorithm for peptide sequencing by mass spectrometry. *Biomed. Environ. Mass Spectrom.*, 19:363–368, 1990.

[10] J. M. S. Bartlett and D. Stirling. A short history of the polymerase chain reaction. *Methods Mol. Biol.*, 226:3–6, 2003.

[11] C. Bauer, R. Cramer and J. Schuchhardt. Evaluation of peak-picking algorithms for protein mass spectrometry. *Methods Mol. Biol.*, 696:341–352, 2011.

[12] M. Beck, I. M. Gessel and T. Komatsu. The polynomial part of a restricted partition function related to the Frobenius problem. *Electron. J. Comb.*, 8(1):N7, 2001.

[13] D. E. Beihoffer, J. Hendry, A. Nijenhuis and S. Wagon. Faster algorithms for Frobenius numbers. *Electron. J. Comb.*, 12:R27, 2005.

[14] C. Benecke, T. Grüner, A. Kerber, R. Laue and T. Wieland. MOLecular Structure GENeration with MOLGEN, new features and future developments. *Anal. Chim. Acta*, 314:141–147, 1995.

[15] G. Benson. Composition alignment. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI 2003)*, volume 2812 of *Lect. Notes Comput. Sc.*, pages 447–461. Springer, 2003.

[16] M. W. Bern and D. Goldberg. EigenMS: De novo analysis of peptide tandem mass spectra by spectral graph partitioning. In *Proc. of Research in Computational Molecular Biology (RECOMB 2005)*, volume 3500 of *Lect. Notes Comput. Sc.*, pages 357–372. Springer, 2005.

[17] M. W. Bern and D. Goldberg. De novo analysis of peptide tandem mass spectra by spectral graph partitioning. *J. Comput. Biol.*, 13(2):364–378, 2006.

[18] A. Bertsch, A. Leinenbach, A. Pervukhin, M. Lubeck, R. Hartmer, C. Baessmann, Y. A. Elnakady, R. Müller, S. Böcker, C. G. Huber, and O. Kohlbacher. De novo peptide sequencing by tandem MS using complementary CID and electron transfer dissociation. *Electrophoresis*, 30(21):3736–3747, 2009.

[19] K. Biemann, C. Cone and B. R. Webster. Computer-aided interpretation of high-resolution mass spectra. II. Amino acid sequence of peptides. *J. Am. Chem. Soc.*, 88(11):2597–2598, 1966.

[20] K. Biemann, C. Cone, B. R. Webster and G. P. Arsenault. Determination of the amino acid sequence in oligopeptides by computer interpretation of their high-resolution mass spectra. *J. Am. Chem. Soc.*, 88(23):5598–5606, 1966.

[21] A. Björklund, T. Husfeldt, P. Kaski and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. of ACM Symposium on Theory of Computing (STOC 2007)*, pages 67–74. ACM Press New York, 2007.

[22] N. Blow. Glycobiology: A spoonful of sugar. *Nature*, 457(7229):617–620, 2009.

[23] S. Böcker. Sequencing from compomers: Using mass spectrometry for DNA de-novo sequencing of 200+ nt. *J. Comput. Biol.*, 11(6):1110–1134, 2004.

[24] S. Böcker and Zs. Lipták. A fast and simple algorithm for the Money Changing Problem. *Algorithmica*, 48(4):413–432, 2007.

[25] S. Böcker and V. Mäkinen. Combinatorial approaches for mass spectra recalibration. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 5(1):91–100, 2008.

[26] S. Böcker and F. Rasche. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinformatics*, 24:I49–I55, 2008. Proc. of *European Conference on Computational Biology* (ECCB 2008).

[27] S. Böcker, M. Letzel, Zs. Lipták and A. Pervukhin. Decomposing metabolomic isotope patterns. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI 2006)*, volume 4175 of *Lect. Notes Comput. Sc.*, pages 12–23. Springer, 2006.

[28] S. Böcker, B. Kehr and F. Rasche. Determination of glycan structure from tandem mass spectra. In *Proc. of Computing and Combinatorics Conference (COCOON 2009)*, volume 5609 of *Lect. Notes Comput. Sc.*, pages 258–267. Springer, 2009.

[29] S. Böcker, M. Letzel, Zs. Lipták and A. Pervukhin. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25(2):218–224, 2009.

[30] S. Böcker, F. Rasche and T. Steijger. Annotating fragmentation patterns. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI 2009)*, volume 5724 of *Lect. Notes Comput. Sc.*, pages 13–24. Springer, 2009.

[31] A. Brauer and J. E. Shockley. On a problem of Frobenius. *J. Reine Angew. Math.*, 211: 215–220, 1962.

[32] R. Breitling, A. R. Pitt and M. P. Barrett. Precision mapping of the metabolome. *Trends Biotechnol.*, 24(12):543–548, 2006.

[33] K. Q. Brown. *Geometric transforms for fast geometric algorithms*. Report cmucs-80-101, Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, USA, 1980.

[34] S. Cappadona, P. Nanni, M. Benevento, F. Levander, P. Versura, A. Roda, S. Cerutti, and L. Pattini. Improved label-free LC-MS analysis by wavelet-based noise rejection. *J Biomed Biotechnol*, 2010:131505, 2010.

[35] A. Ceroni, K. Maass, H. Geyer, R. Geyer, A. Dell and S. M. Haslam. GlycoWorkbench: a tool for the computer-assisted annotation of mass spectra of glycans. *J. Proteome Res.*, 7 (4):1650–1659, 2008.

[36] D. C. Chamrad, G. Körting, K. Stühler, H. E. Meyer, J. Klose and M. Blüggel. Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data. *Proteomics*, 4:619–628, 2004.

[37] S. Chattopadhyay and P. Das. The K-dense corridor problems. *Pattern Recogn. Lett.*, 11 (7):463–469, 1990.

[38] E. Check. Proteomics and cancer: Running before we can walk? *Nature*, 429:496–497, 2004.

[39] T. Chen, M.-Y. Kao, M. Tepel, J. Rush and G. M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *J. Comput. Biol.*, 8(3):325–337, 2001. Preliminary version in *Proc. of Symposium on Discrete Algorithms (SODA 2000)*, Association for Computing Machinery, 2000, 389–398.

[40] W. L. Chen. Chemoinformatics: past, present, and future. *J. Chem. Inf. Model.*, 46(6): 2230–2255, 2006.

[41] F. Y. Chin, C. A. Wang and F. L. Wang. Maximum stabbing line in 2D plane. In *Proc. of Conf. on Computing and Combinatorics (COCOON 1999)*, volume 1627 of *Lect. Notes Comput. Sc.*, pages 379–388. Springer, 1999.

[42] H. H. Chou, H. Takematsu, S. Diaz, J. Iber, E. Nickerson, K. L. Wright, E. A. Muchmore, D. L. Nelson, S. T. Warren, and A. Varki. A mutation in human CMP-sialic acid hydroxylase occurred after the Homo-Pan divergence. *Proc. Natl. Acad. Sci. U. S. A.*, 95(20):11751–11756, 1998.

[43] Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Sci. Sinica*, 14: 1396–1400, 1965.

[44] K. R. Clauser, P. Baker and A. L. Burlingame. Role of accurate mass measurement (+/− 10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Anal. Chem.*, 71(14):2871–2882, 1999.

[45] C. A. Cooper, E. Gasteiger and N. H. Packer. GlycoMod – a software tool for determining glycosylation compositions from mass spectrometric data. *Proteomics*, 1(2):340–349, 2001.

[46] C. A. Cooper, H. J. Joshi, M. J. Harrison, M. R. Wilkins and N. H. Packer. GlycoSuiteDB: a curated relational database of glycoprotein glycan structures and their biological sources. 2003 update. *Nucleic Acids Res.*, 31(1):511–513, 2003.

[47] R. Craig and R. C. Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.

[48] V. Dančík, T. A. Addona, K. R. Clauser, J. E. Vath and P. A. Pevzner. De novo peptide sequencing via tandem mass spectrometry: A graph-theoretical approach. *J. Comput. Biol.*, 6(3/4):327–342, 1999. Preliminary version in *Proc. of Research in Computational Molecular Biology (RECOMB 1999)*, 135–144.

[49] C. Dass. *Principles and practice of biological mass spectrometry*. John Wiley and Sons, 2001.

[50] R. Datta and M. W. Bern. Spectrum fusion: using multiple mass spectra for de novo peptide sequencing. *J. Comput. Biol.*, 16(8):1169–1182, 2009.

[51] J. L. Davison. On the linear diophantine problem of Frobenius. *J. Number Theory*, 48(3): 353–363, 1994.

[52] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, second edition, 2000.

[53] E. de Hoffmann and V. Stroobant. *Mass Spectrometry: Principles and Applications*. Wiley-Interscience, third edition, 2007.

[54] J. R. de Laeter, J. K. Böhlke, P. D. Biévre, H. Hidaka, H. S. Peiser, K. J. R. Rosman and P. D. P. Taylor. Atomic weights of the elements. Review 2000 (IUPAC technical report). *Pure Appl. Chem.*, 75(6):683–800, 2003.

[55] E. W. Deutsch, H. Lam and R. Aebersold. Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics. *Physiological Genomics*, 33:18–25, 2008.

[56] P. A. DiMaggio and C. A. Floudas. De novo peptide identification via tandem mass spectrometry and integer linear optimization. *Anal. Chem.*, 79(4):1433–1446, 2007.

[57] B. Domon and R. Aebersold. Mass spectrometry and protein analysis. *Science*, 312:212–217, 2006.

[58] B. Domon and C. E. Costello. A systematic nomenclature for carbohydrate fragmentations in FAB-MS/MS spectra of glycoconjugates. *Glycoconjugate J.*, 5:397–409, 1988.

[59] R. Dondi, G. Fertin and S. Vialette. Complexity issues in vertex-colored graph pattern matching. *J. Discrete Algorithms*, 9:82–99, 2011.

[60] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[61] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1972.

[62] M. Dyer. Approximate counting by dynamic programming. In *Proc. of Symposium on Theory of Computing (STOC 2003)*, pages 693–699, 2003.

[63] S. R. Eddy. "antedisciplinary" science. *PLoS Comput. Biol.*, 1(1):e6, 2005.

[64] P. Edman. Method for determination of the amino acid sequence in peptides. *Acta Chem. Scand.*, 4:283–293, 1950.

[65] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Stand.*, 71B:233–240, 1967.

[66] M. Ehrich, S. Böcker and D. van den Boom. Multiplexed discovery of sequence polymorphisms using base-specific cleavage and MALDI-TOF MS. *Nucleic Acids Res.*, 33(4):e38, 2005.

[67] D. Einstein, D. Lichtblau, A. Strzebonski and S. Wagon. Frobenius numbers by lattice point enumeration. *INTEGERS*, 7(1):#A15, 2007.

[68] J. E. Elias and S. P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*, 4(3):207–214, 2007.

[69] J. E. Elias, F. D. Gibbons, O. D. King, F. P. Roth and S. P. Gygi. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. Biotechnol.*, 22(2):214–219, 2004.

[70] J. K. Eng, A. L. McCormack and J. R. Yates III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectr.*, 5:976–989, 1994.

[71] M. Ethier, J. A. Saba, M. Spearman, O. Krokhin, M. Butler, W. Ens, K. G. Standing, and H. Perreault. Application of the StrOligo algorithm for the automated structure assignment of complex N-linked glycans from glycoproteins using tandem mass spectrometry. *Rapid Commun. Mass Spectrom.*, 17(24):2713–2720, 2003.

[72] M. Fellows, G. Fertin, D. Hermelin and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP 2007)*, volume 4596 of *Lect. Notes Comput. Sc.*, pages 340–351. Springer, 2007.

[73] J. Fenn, M. Mann, C. Meng, S. Wong and C. Whitehouse. Electrospray ionisation for mass spectrometry of large biomolecules. *Science*, 246:64–71, 1989.

[74] D. Fenyö and R. C. Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.*, 75(4):768–774, 2003.

[75] J. Fernández-de-Cossío, L. J. Gonzalez and V. Besada. A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *Comput. Appl. Biosci.*, 11(4):427–434, 1995.

[76] J. Fernández-de-Cossío, J. Gonzalez, T. Takao, Y. Shimonishi, G. Padron and V. Besada. A software program for the rapid sequence analysis of unknown peptides involving modifications, based on MS/MS data. In *ASMS Conf. on Mass Spectrometry and Allied Topics, Slot 074*, 1997.

[77] J. Fernández-de-Cossío, L. J. Gonzalez, Y. Satomi, L. Betancourt, Y. Ramos, V. Huerta, A. Amaro, V. Besada, G. Padron, N. Minamino, and T. Takao. Isotopica: a tool for the calculation and viewing of complex isotopic envelopes. *Nucleic Acids Res.*, 32(Web Server issue):W674–W678, 2004.

[78] A. R. Fernie, R. N. Trethewey, A. J. Krotzky and L. Willmitzer. Metabolite profiling: from diagnostics to systems biology. *Nat. Rev. Mol. Cell Biol.*, 5(9):763–769, 2004.

[79] H. I. Field, D. Fenyö and R. C. Beavis. RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. *Proteomics*, 2(1):36–47, 2002.

[80] B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J. M. Buhmann. NovoHMM: a hidden Markov model for de novo peptide sequencing. *Anal. Chem.*, 77(22):7265–7273, 2005.

[81] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. Freely available from `http://algo.inria.fr/flajolet/Publications/book.pdf`.

[82] A. Frank and P. Pevzner. PepNovo: de novo peptide sequencing via probabilistic network modeling. *Anal. Chem.*, 15:964–973, 2005.

[83] A. M. Frank, M. M. Savitski, M. N. Nielsen, R. A. Zubarev and P. A. Pevzner. De novo peptide sequencing and identification with precision mass spectrometry. *J. Proteome Res.*, 6(1):114–123, 2007.

[84] 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.

[85] V. A. Fusaro, D. R. Mani, J. P. Mesirov and S. A. Carr. Prediction of high-responding peptides for targeted protein assays by mass spectrometry. *Nat. Biotechnol.*, 27(2):190–198, 2009.

[86] H. Gabow, Z. Galil, T. Spencer and R. Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6:109–122, 1986.

[87] M. R. Garey and D. S. Johnson. *Computers and Intractability (A Guide to Theory of NP-Completeness)*. Freeman, New York, 1979.

[88] J. Gasteiger, W. Hanebeck and K.-P. Schulz. Prediction of mass spectra from structural information. *J. Chem. Inf. Comput. Sci.*, 32(4):264–271, 1992.

[89] S. P. Gaucher, J. Morrow and J. A. Leary. STAT: a saccharide topology analysis tool used in combination with tandem mass spectrometry. *Anal. Chem.*, 72(11):2331–2336, 2000.

[90] L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi, and S. H. Bryant. Open mass spectrometry search algorithm. *J. Proteome Res.*, 3:958–964, 2004.

[91] P. Gilmore and R. Gomory. Multi-stage cutting stock problems of two and more dimensions. *Oper. Res.*, 13(1):94–120, 1965.

[92] D. Goldberg, M. Sutton-Smith, J. Paulson and A. Dell. Automatic annotation of matrix-assisted laser desorption/ionization N-glycan spectra. *Proteomics*, 5(4):865–875, 2005.

[93] D. Goldberg, M. W. Bern, B. Li and C. B. Lebrilla. Automatic determination of O-glycan structure from fragmentation spectra. *J. Proteome Res.*, 5(6):1429–1434, 2006.

[94] D. Goldberg, M. W. Bern, S. Parry, M. Sutton-Smith, M. Panico, H. R. Morris and A. Dell. Automated N-glycopeptide identification using a combination of single- and tandem-MS. *J. Proteome Res.*, 6(10):3995–4005, 2007.

[95] D. Goldberg, M. W. Bern, S. J. North, S. M. Haslam and A. Dell. Glycan family analysis for deducing N-glycan topology from single MS. *Bioinformatics*, 25(3):365–371, 2009.

[96] 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.

[97] N. A. Gray. Applications of artificial intelligence for organic chemistry: Analysis of C-13 spectra. *Artificial Intelligence*, 22(1):1–21, 1984.

[98] N. A. B. Gray, R. E. Carhart, A. Lavanchy, D. H. Smith, T. Varkony, B. G. Buchanan, W. C. White, and L. Creary. Computerized mass spectrum prediction and ranking. *Anal. Chem.*, 52(7):1095–1102, 1980.

[99] N. A. B. Gray, A. Buchs, D. H. Smith and C. Djerassi. Computer assisted structural interpretation of mass spectral data. *Helv. Chim. Acta*, 64(2):458–470, 1981.

[100] H. Greenberg. Solution to a linear diophantine equation for nonnegative integers. *J. Algorithms*, 9(3):343–353, 1988.

[101] D. H. Greene and D. E. Knuth. *Mathematics for the Analysis of Algorithms*, volume 1 of *Progress in Computer Science and Applied Logic (PCS)*. Birkhäuser Boston, 1990.

[102] J. Gross. *Mass Spectrometry: A textbook*. Springer, Berlin, 2004.

[103] K. Grützmann, S. Böcker and S. Schuster. Combinatorics of aliphatic amino acids. *Naturwissenschaften*, 98(1):79–86, 2011.

[104] M. Guilhaus. Principles and instrumentation in time-of-flight mass spectrometry. *J. Mass Spectrom.*, 30:1519–1532, 1995.

[105] S. Guillemot and F. Sikora. Finding and counting vertex-colored subtrees. In *Proc. of Symposium on Mathematical Foundations of Computer Science (MFCS 2010)*, volume 6281 of *Lect. Notes Comput. Sc.*, pages 405–416. Springer, 2010.

[106] C. Hamm, W. Wilson and D. Harvan. Peptide sequencing program. *Comput. Appl. Biosci.*, 2:115–118, 1986.

[107] F. Harary, R. W. Robinson and A. J. Schwenk. Twenty-step algorithm for determining the asymptotic number of trees of various species. *J. Austral. Math. Soc.*, 20(Series A): 483–503, 1975.

[108] M. Havilio, Y. Haddad and Z. Smilansky. Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem.*, 75:435–444, 2003.

[109] M. Heinonen, A. Rantanen, T. Mielikäinen, J. Kokkonen, J. Kiuru, R. A. Ketola and J. Rousu. FiD: a software for ab initio structural identification of product ions from tandem mass spectrometric data. *Rapid Commun. Mass Spectrom.*, 22(19):3043–3052, 2008.

[110] D. W. Hill, T. M. Kertesz, D. Fontaine, R. Friedman and D. F. Grant. Mass spectral metabonomics beyond elemental formula: Chemical database querying by matching experimental with computational fragmentation spectra. *Anal. Chem.*, 80(14):5574–5582, 2008.

[111] W. M. Hines, A. M. Falick, A. L. Burlingame and B. W. Gibson. Pattern-based algorithm for peptide sequencing from tandem high energy collision-induced dissociation mass spectra. *J. Am. Soc. Mass Spectrom.*, 3(4):326 – 336, 1992.

[112] C. A. R. Hoare. FIND (algorithm 65). *Communications of the ACM*, 4:321–322, 1961.

[113] D. H. Horn, R. A. Zubarev and F. W. McLafferty. Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *J. Am. Soc. Mass Spectr.*, 11:320–332, 2000.

[114] C. S. Hsu. Diophantine approach to isotopic abundance calculations. *Anal. Chem.*, 56(8): 1356–1361, 1984.

[115] Q. Hu, R. J. Noll, H. Li, A. Makarov, M. Hardman and R. G. Cooks. The Orbitrap: a new mass spectrometer. *J. Mass Spectrom.*, 40(4):430–443, 2005.

[116] R. Hussong and A. Hildebrandt. Signal processing in proteomics. *Methods Mol. Biol.*, 604: 145–161, 2010.

[117] N. Jaitly, M. E. Monroe, V. A. Petyuk, T. R. W. Clauss, J. N. Adkins and R. D. Smith. Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline. *Anal. Chem.*, 78(21):7397–7409, 2006.

[118] N. Jeffries. Algorithms for alignment of mass spectrometry proteomic data. *Bioinformatics*, 21(14):3066–3073, 2005.

[119] R. S. Johnson and J. A. Taylor. Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry. *Methods Mol. Biol.*, 146:41–61, 2000.

[120] R. S. Johnson and J. A. Taylor. Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry. *Mol. Biotechnol.*, 22(3):301–315, 2002.

[121] P. Jones, R. G. Côté, L. Martens, A. F. Quinn, C. F. Taylor, W. Derache, H. Hermjakob, and R. Apweiler. PRIDE: a public repository of protein and peptide identifications for the proteomics community. *Nucleic Acids Res.*, 34(Database-Issue):659–663, 2006.

[122] H. J. Joshi, M. J. Harrison, B. L. Schulz, C. A. Cooper, N. H. Packer and N. G. Karlsson. Development of a mass fingerprinting tool for automated interpretation of oligosaccharide fragmentation data. *Proteomics*, 4(6):1650–1664, 2004.

[123] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble and M. J. MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods*, 4(11): 923–925, 2007.

[124] 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. *Nucleic Acids Res.*, 34:D354–D357, 2006.

[125] R. Kannan. Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12:161–177, 1991.

[126] E. A. Kapp, F. Schütz, L. M. Connolly, J. A. Chakel, J. E. Meza, C. A. Miller, D. Fenyo, J. K. Eng, J. N. Adkins, G. S. Omenn, and R. J. Simpson. An evaluation, comparison, and accurate benchmarking of several publicly available MS/MS search algorithms: Sensitivity and specificity analysis. *Proteomics*, 5:3475–3490, 2005.

[127] M. Karas and F. Hillenkamp. Laser desorption ionization of proteins with molecular masses exceeding 10,000 Daltons. *Anal. Chem.*, 60:2299–2301, 1988.

[128] A. Keller, A. I. Nesvizhskii, E. Kolker and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.*, 74(20):5383–5392, 2002.

[129] A. Keller, J. Eng, N. Zhang, X.-J. Li and R. Aebersold. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.*, 1:2005.0017, 2005.

[130] E. Kendrick. A mass scale based on CH2 = 14.0000 for high resolution mass spectrometry of organic compounds. *Anal. Chem.*, 35(13):2146–2154, 1963.

[131] A. Kerber, R. Laue and D. Moser. Ein Strukturgenerator für molekulare Graphen. *Anal. Chim. Acta*, 235:221 – 228, 1990.

[132] A. Kerber, R. Laue, M. Meringer and C. Rücker. Molecules in silico: The generation of structural formulae and its applications. *J. Comput. Chem. Japan*, 3(3):85–96, 2004.

[133] S. Kim, N. Gupta and P. A. Pevzner. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.*, 7(8):3354–3363, 2008.

[134] S. Kim, N. Bandeira and P. A. Pevzner. Spectral profiles, a novel representation of tandem mass spectra and their applications for de novo peptide sequencing and identification. *Mol. Cell. Proteomics*, 8(6):1391–1400, 2009.

[135] S. Kim, N. Gupta, N. Bandeira and P. A. Pevzner. Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. *Mol. Cell. Proteomics*, 8(1):53–69, 2009.

[136] 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, 2006.

[137] T. Kind and O. Fiehn. Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*, 8:105, 2007.

[138] H. Kubinyi. Calculation of isotope distributions in mass spectrometry: A trivial solution for a non-trivial problem. *Anal. Chim. Acta*, 247:107–119, 1991.

[139] K.-S. Kwok, R. Venkataraghavan and F. W. McLafferty. Computer-aided interpretation of mass spectra. III. Self-training interpretive and retrieval system. *J. Am. Chem. Soc.*, 95 (13):4185–4194, 1973.

[140] V. Lacroix, C. G. Fernandes, and M.-F. Sagot. Motif search in graphs: Application to metabolic networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 3(4):360–368, 2006.

[141] A. J. Lapadula, P. J. Hatcher, A. J. Hanneman, D. J. Ashline, H. Zhang and V. N. Reinhold. Congruent strategies for carbohydrate sequencing. 3. OSCAR: an algorithm for assigning oligosaccharide topology from MS^n data. *Anal. Chem.*, 77(19):6271–6279, 2005.

[142] R. L. Last, A. D. Jones and Y. Shachar-Hill. Towards the plant metabolome and beyond. *Nat. Rev. Mol. Cell Biol.*, 8:167–174, 2007.

[143] A. Lavanchy, T. Varkony, D. H. Smith, N. A. B. Gray, W. C. White, R. E. Carhart, B. G. Buchanan, and C. Djerassi. Rule-based mass spectrum prediction and ranking: Applications to structure elucidation of novel marine sterols. *Org. Mass Spectrom.*, 15(7): 355–366, 1980.

[144] J. Lederberg. Topological mapping of organic molecules. *Proc. Natl. Acad. Sci. U. S. A.*, 53 (1):134–139, 1965.

[145] J. Lederberg. How DENDRAL was conceived and born. In *ACM Conference on the History of Medical Informatics, History of Medical Informatics archive*, pages 5–19, 1987. Available from `http://doi.acm.org/10.1145/41526.41528`.

[146] T. A. Lee. *A Beginner's Guide to Mass Spectral Interpretation*. Wiley, 1998.

[147] M. Lefmann, C. Honisch, S. Boecker, N. Storm, F. von Wintzingerode, C. Schloetelburg, A. Moter, D. van den Boom, and U. B. Goebel. A novel mass spectrometry based tool for genotypic identification of mycobacteria. *J. Clin. Microbiol.*, 42(1):339–346, 2004.

[148] G. Li and F. Ruskey. The advantages of forward thinking in generating rooted and free trees. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA 1999)*, pages 939–940, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[149] G. Liu, J. Zhang, B. Larsen, C. Stark, A. Breitkreutz, Z.-Y. Lin, B.-J. Breitkreutz, Y. Ding, K. Colwill, A. Pasculescu, T. Pawson, J. L. Wrana, A. I. Nesvizhskii, B. Raught, M. Tyers, and A.-C. Gingras. ProHits: integrated software for mass spectrometry-based interaction proteomics. *Nat. Biotechnol.*, 28(10):1015–1017, 2010.

[150] K. K. Lohmann and C.-W. von der Lieth. GlycoFragment and GlycoSearchMS: web tools to support the interpretation of mass spectra of complex carbohydrates. *Nucleic Acids Res.*, 32(Web Server issue):W261–W266, 2004.

[151] B. Lu and T. Chen. A suffix tree approach to the interpretation of tandem mass spectra: Applications to peptides of non-specific digestion and post-translational modifications. *Bioinformatics*, 19(Suppl 2):ii113–ii121, 2003. Proc. of *European Conference on Computational Biology* (ECCB 2003).

[152] A. Luedemann, K. Strassburg, A. Erban and J. Kopka. TagFinder for the quantitative analysis of gas chromatography–mass spectrometry (GC-MS)-based metabolite profiling experiments. *Bioinformatics*, 24(5):732–737, 2008.

[153] G. S. Lueker. Two NP-complete problems in nonnegative integer programming. Technical Report TR-178, Department of Electrical Engineering, Princeton University, 1975.

[154] Y.-R. Luo. *Handbook of Bond Dissociation Energies in Organic Compounds*. CRC Press, Boca Raton, 2003.

[155] B. Ma and G. Lajoie. Improving the de novo sequencing accuracy by combining two independent scoring functions in peaks software. Poster at the ASMS Conference on Mass Spectrometry and Allied Topics, 2005.

[156] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby and G. Lajoie. PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Commun. Mass Spectrom.*, 17(20):2337–2342, 2003.

[157] B. Ma, K. Zhang and C. Liang. An effective algorithm for peptide de novo sequencing from MS/MS spectra. *J. Comput. Syst. Sci.*, 70:418–430, 2005.

[158] K. Maass, R. Ranzinger, H. Geyer, C.-W. von der Lieth and R. Geyer. "Glyco-peakfinder" – de novo composition analysis of glycoconjugates. *Proteomics*, 7(24):4435–4444, 2007.

[159] P. Mallick, M. Schirle, S. S. Chen, M. R. Flory, H. Lee, D. Martin, J. Ranish, B. Raught, R. Schmitt, T. Werner, B. Kuster, and R. Aebersold. Computational prediction of proteotypic peptides for quantitative proteomics. *Nat. Biotechnol.*, 25(1):125–131, 2007.

[160] M. Mann and M. Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal. Chem.*, 66(24):4390–4399, 1994.

[161] S. Martello and P. Toth. An exact algorithm for large unbounded knapsack problems. *Oper. Res. Lett.*, 9(1):15–20, 1990.

[162] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, 1990.

[163] R. Matthiesen, J. Bunkenborg, A. Stensballe, O. N. Jensen, K. G. Welinder and G. Bauw. Database-independent, database-dependent, and extended interpretation of peptide mass spectra in VEMS V2.0. *Proteomics*, 4(9):2583–2593, 2004.

[164] R. Matthiesen, M. B. Trelle, P. Hojrup, J. Bunkenborg and O. N. Jensen. VEMS 3.0: algorithms and computational tools for tandem mass spectrometry based identification of post-translational modifications in proteins. *J. Proteome Res.*, 4(6):2338–2347, 2005.

[165] L. McHugh and J. W. Arthur. Computational methods for protein identification from mass spectrometry data. *PLoS Comput. Biol.*, 4(2):e12, 2008.

[166] P. E. Miller and M. B. Denton. The quadrupole mass filter: Basic operating concepts. *J. Chem. Educ.*, 63:617–622, 1986.

[167] L. Mo, D. Dutta, Y. Wan and T. Chen. MSNovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry. *Anal. Chem.*, 79(13):4870–4878, 2007.

[168] E. Mostacci, C. Truntzer, H. Cardot and P. Ducoroy. Multivariate denoising methods combining wavelets and principal component analysis for mass spectrometry data. *Proteomics*, 10(14):2564–2572, 2010.

[169] I. K. Mun and F. W. Mclafferty. Computer methods of molecular structure elucidation from unknown mass spectra. In *Supercomputers in Chemistry*, ACS Symposium Series, chapter 9, pages 117–124. American Chemical Society, 1981.

[170] S. Na, J. Jeong, H. Park, K.-J. Lee and E. Paek. Unrestrictive identification of multiple post-translational modifications from tandem mass spectrometry using an error-tolerant algorithm based on an extended sequence tag approach. *Mol. Cell. Proteomics*, 7(12): 2452–2463, 2008.

[171] S. Neumann and S. Böcker. Computational mass spectrometry for metabolomics – a review. *Anal. Bioanal. Chem.*, 398(7):2779–2788, 2010.

[172] N. Nguyen, H. Huang, S. Oraintara and A. Vo. Mass spectrometry data processing using zero-crossing lines in multi-scale of Gaussian derivative wavelet. *Bioinformatics*, 26(18): i659–i665, 2010.

[173] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[174] J. A. November. *Digitizing life: the introduction of computers to biology and medicine*. PhD thesis, Princeton University, Princeton, USA, 2006.

[175] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and inter-laboratory transferability of a tandem mass spectral reference library: 1. results of an austrian multicenter study. *J. Mass Spectrom.*, 44(4):485–493, 2009.

[176] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and the inter-laboratory transferability of a tandem mass spectral reference library: 2. optimization and characterization of the search algorithm. *J. Mass Spectrom.*, 44(4):494–502, 2009.

[177] S. Orchard, L. Montechi-Palazzi, E. W. Deutsch, P.-A. Binz, A. R. Jones, N. Paton, A. Pizarro, D. M. Creasy, J. Wojcik, and H. Hermjakob. Five years of progress in the standardization of proteomics data: 4th annual spring workshop of the HUPO-proteomics standards initiative. *Proteomics*, 7:3436–3440, 2007.

[178] R. Otter. The number of trees. *The Annals of Mathematics*, 49(3):583–599, 1948.

[179] K. G. Owens. Application of correlation analysis techniques to mass spectral data. *Appl. Spectrosc. Rev.*, 27(1):1–49, 1992.

[180] N. H. Packer, C.-W. von der Lieth, K. F. Aoki-Kinoshita, C. B. Lebrilla, J. C. Paulson, R. Raman, P. Rudd, R. Sasisekharan, N. Taniguchi, and W. S. York. Frontiers in glycomics: bioinformatics and biomarkers in disease. An NIH white paper prepared from discussions by the focus groups at a workshop on the NIH campus, Bethesda MD (September 11-13, 2006). *Proteomics*, 8(1):8–20, 2008.

[181] G. Palmisano, D. Antonacci and M. R. Larsen. Glycoproteomic profile in wine: a 'sweet' molecular renaissance. *J. Proteome Res.*, 9(12):6148–6159, 2010.

[182] D. J. Pappin, P. Hojrup and A. Bleasby. Rapid identification of proteins by peptide-mass fingerprinting. *Curr. Biol.*, 3(6):327–332, 1993.

[183] C. Y. Park, A. A. Klammer, L. Käll, M. J. MacCoss and W. S. Noble. Rapid and accurate peptide identification from tandem mass spectra. *J. Proteome Res.*, 7(7):3022–3027, 2008.

[184] W. E. Parkins. The uranium bomb, the calutron, and the space-charge problem. *Physics Today*, 58(5):45–51, 2005.

[185] V. Pellegrin. Molecular formulas of organic compounds: the nitrogen rule and degree of unsaturation. *J. Chem. Educ.*, 60(8):626–633, 1983.

[186] D. N. Perkins, D. J. Pappin, D. M. Creasy and J. S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[187] R. H. Perry, R. G. Cooks and R. J. Noll. Orbitrap mass spectrometry: instrumentation, ion motion and applications. *Mass Spectrom. Rev.*, 27(6):661–699, 2008.

[188] G. Pólya. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta Mathematica*, 68(1):145–254, 1937.

[189] S. C. Pomerantz, J. A. Kowalak and J. A. McCloskey. Determination of oligonucleotide composition from mass spectromectrically measured molecular weight. *J. Am. Soc. Mass Spectrom.*, 4:204–209, 1993.

[190] R. Raman, S. Raguram, G. Venkataraman, J. C. Paulson and R. Sasisekharan. Glycomics: an integrated systems approach to structure-function relationships of glycans. *Nat. Methods*, 2(11):817–824, 2005.

[191] J. L. Ramírez-Alfonsín. *The Diophantine Frobenius Problem*. Oxford University Press, 2005.

[192] J. L. Ramírez-Alfonsín. Complexity of the Frobenius problem. *Combinatorica*, 16(1):143–147, 1996.

[193] I. Rauf, F. Rasche and S. Böcker. Computing maximum colorful subtrees in practice. Manuscript. **[TODO: REMOVE OR UPDATE]**, 2011.

[194] A. L. Rockwood and P. Haimi. Efficient calculation of accurate masses of isotopic peaks. *J. Am. Soc. Mass Spectrom.*, 17(3):415–419, 2006.

[195] A. L. Rockwood, M. M. Kushnir and G. J. Nelson. Dissociation of individual isotopic peaks: Predicting isotopic distributions of product ions in MS^n. *J. Am. Soc. Mass Spectr.*, 14:311–322, 2003.

[196] 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.

[197] P. Roepstorff and J. Fohlman. Proposal for a common nomenclature for sequence ions in mass spectra of peptides. *Biomed. Mass Spectrom.*, 11(11):601, 1984.

[198] S. Rogers, R. A. Scheltema, M. Girolami and R. Breitling. Probabilistic assignment of formulas to mass peaks in metabolomics experiments. *Bioinformatics*, 25(4):512–518, 2009.

[199] R. G. Sadygov and J. R. Yates III. A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal. Chem.*, 75(15):3792–3798, 2003.

[200] R. G. Sadygov, D. Cociorva and J. R. Yates III. Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nat. Methods*, 1(3): 195–202, 2004.

[201] T. Sakurai, T. Matsuo, H. Matsuda and I. Katakuse. PAAS 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biomed. Mass Spectrom.*, 11(8):396–399, 1984.

[202] A. Salomaa. Counting (scattered) subwords. *B. Euro. Assoc. Theo. Comp. Sci.*, 81:165–179, 2003.

[203] F. Sanger, S. Nicklen and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. U.S.A.*, 74(12):5463–5467, 1977.

[204] M. M. Savitski, M. L. Nielsen, F. Kjeldsen and R. A. Zubarev. Proteomics-grade de novo sequencing approach. *J. Proteome Res.*, 4:2348–2354, 2005.

[205] K. Scheubert, F. Hufsky, F. Rasche and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. In *Proc. of Research in Computational Molecular Biology (RECOMB 2011)*, volume 6577 of *Lect. Notes Comput. Sc.*, pages 377–391. Springer, 2011.

[206] J. Seidler, N. Zinn, M. E. Boehm and W. D. Lehmann. De novo sequencing of peptides by MS/MS. *Proteomics*, 10(4):634–649, 2010.

[207] J. Senior. Partitions and their representative graphs. *Amer. J. Math.*, 73(3):663–689, 1951.

[208] B. Shan, B. Ma, K. Zhang and G. Lajoie. Complexities and algorithms for glycan sequencing using tandem mass spectrometry. *J. Bioinformatics and Computational Biology*, 6(1): 77–91, 2008.

[209] Q. Sheng, Y. Mechref, Y. Li, M. V. Novotny and H. Tang. A computational approach to characterizing bond linkages of glycan isomers using matrix-assisted laser desorption/ionization tandem time-of-flight mass spectrometry. *Rapid Commun. Mass Spectrom.*, 22(22):3561–3569, 2008.

[210] I. V. Shilov, S. L. Seymour, A. A. Patel, A. Loboda, W. H. Tang, S. P. Keating, C. L. Hunter, L. M. Nuwaysir, and D. A. Schaeffer. The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Mol. Cell. Proteomics*, 6(9):1638–1655, 2007.

[211] H. Shin, M. P. Sampat, J. M. Koomen and M. K. Markey. Wavelet-based adaptive denoising and baseline correction for MALDI TOF MS. *OMICS*, 14(3):283–295, 2010.

[212] F. Sikora. An (almost complete) state of the art around the graph motif problem. Technical report, Université Paris-Est, France, 2010. Available from `http://www-igm.univ-mlv.fr/~fsikora/pub/GraphMotif-Resume.pdf`.

[213] R. M. Silverstein, F. X. Webster and D. Kiemle. *Spectrometric Identification of Organic Compounds*. Wiley, 7th edition, 2005.

[214] G. Siuzdak. *The Expanding Role of Mass Spectrometry in Biotechnology*. MCC Press, second edition, 2006.

[215] D. H. Smith, N. A. Gray, J. G. Nourse and C. W. Crandell. The DENDRAL project: recent advances in computer-assisted structure elucidation. *Anal. Chim. Acta*, 133(4):471 – 497, 1981.

[216] R. K. Snider. Efficient calculation of exact mass isotopic distributions. *J. Am. Soc. Mass Spectrom.*, 18(8):1511–1515, 2007.

[217] H. M. Sobell. Actinomycin and DNA transcription. *Proc. Natl. Acad. Sci. U. S. A.*, 82(16): 5328–5331, 1985.

[218] H. Steen and M. Mann. The ABC's (and XYZ's) of peptide sequencing. *Nature Rev.*, 5: 699–711, 2004.

[219] M. T. Sykes and J. R. Williamson. Envelope: interactive software for modeling and fitting complex isotope distributions. *BMC Bioinformatics*, 9:446, 2008.

[220] J. J. Sylvester and W. J. Curran Sharp. Problem 7382. *Educational Times*, 37:26, 1884.

[221] D. L. Tabb, M. J. MacCoss, C. C. Wu, S. D. Anderson and J. R. Yates. Similarity among tandem mass spectra from proteomic experiments: detection, significance, and utility. *Anal. Chem.*, 75(10):2470–2477, 2003.

[222] H. Tang, Y. Mechref and M. V. Novotny. Automated interpretation of MS/MS spectra of oligosaccharides. *Bioinformatics*, 21 Suppl 1:i431–i439, 2005. Proc. of *Intelligent Systems for Molecular Biology* (ISMB 2005).

[223] S. Tanner, H. Shu, A. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna. Inspect: Identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.*, 77:4626–4639, 2005.

[224] J. A. Taylor and R. S. Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal. Chem.*, 73(11):2594–2604, 2001.

[225] J. A. Taylor and R. S. Johnson. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid Commun. Mass Spectrom.*, 11:1067–1075, 1997.

[226] J. van Lint and R. Wilson. *A Course in Combinatorics*. Cambridge University Press, 2001.

[227] A. Varki, R. D. Cummings, J. D. Esko, H. H. Freeze, P. Stanley, C. R. Bertozzi, G. W. Hart, and M. E. Etzler, editors. *Essentials of Glycobiology*. Cold Spring Harbor Laboratory Press, second edition, 2009. Freely available from `http://www.ncbi.nlm.nih.gov/books/NBK1908/`.

[228] R. Venkataraghavan, F. W. McLafferty and G. E. van Lear. Computer-aided interpretation of mass spectra. *Org. Mass Spectrom.*, 2(1):1–15, 1969.

[229] C.-W. von der Lieth, A. Bohne-Lang, K. K. Lohmann and M. Frank. Bioinformatics for glycomics: status, methods, requirements and perspectives. *Brief. Bioinform.*, 5(2):164–178, 2004.

[230] S. A. Waksman and H. B. Woodruff. Bacteriostatic and bacteriocidal substances produced by soil actinomycetes. *Proc. Soc. Exper. Biol.*, 45:609–614, 1940.

[231] M. S. Waterman and M. Vingron. Rapid and accurate estimates of statistical significance for sequence data base searches. *Proc. Natl. Acad. Sci. U. S. A.*, 91(11):4625–4628, 1994.

[232] J. T. Watson and O. D. Sparkman. *Introduction to Mass Spectrometry: Instrumentation, Applications, and Strategies for Data Interpretation*. Wiley, 2007.

[233] M. E. Wieser. Atomic weights of the elements 2005 (IUPAC technical report). *Pure Appl. Chem.*, 78(11):2051–2066, 2006.

[234] H. Wilf. *generatingfunctionology*. Academic Press, second edition, 1994. Freely available from `http://www.math.upenn.edu/~wilf/DownldGF.html`.

[235] S. Wolf, S. Schmidt, M. Müller-Hannemann and S. Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinformatics*, 11:148, 2010.

[236] W. E. Wolski, M. Lalowski, P. Jungblut and K. Reinert. Calibration of mass spectrometric peptide mass fingerprint data without specific external or internal calibrants. *BMC Bioinformatics*, 6:203, 2005.

[237] J. W. Wong, G. Cagney and H. M. Cartwright. SpecAlign—processing and alignment of mass spectra datasets. *Bioinformatics*, 21(9):2088–2090, 2005.

[238] L.-C. Wu, H.-H. Chen, J.-T. Horng, C. Lin, N. E. Huang, Y.-C. Cheng and K.-F. Cheng. A novel preprocessing method using Hilbert Huang transform for MALDI-TOF and SELDI-TOF mass spectrometry data. *PLoS One*, 5(8):e12493, 2010.

[239] Y. Wu, Y. Mechref, I. Klouckova, M. V. Novotny and H. Tang. A computational approach for the identification of site-specific protein glycosylations through ion-trap mass spectrometry. In *Proc. of RECOMB 2006 satellite workshop on Systems biology and computational proteomics*, volume 4532 of *Lect. Notes Comput. Sc.*, pages 96–107. Springer, 2007.

[240] C. Xu and B. Ma. Complexity and scoring function of MS/MS peptide de novo sequencing. In *Proc. of Computational Systems Bioinformatics Conference (CSB 2006)*, volume 4 of *Series on Advances in Bioinformatics and Computational Biology*, pages 361–369. Imperial College Press, 2006.

[241] J. Yates, P. Griffin, L. Hood and J. Zhou. Computer aided interpretation of low energy MS/MS mass spectra of peptides. In J. Villafranca, editor, *Techniques in Protein Chemistry II*, pages 477–485. Academic Press, San Diego, 1991.

[242] 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.

[243] J. Zaia. Mass spectrometry of oligosaccharides. *Mass Spectrom. Rev.*, 23(3):161–227, 2004.

[244] J. Zhang, E. Gonzalez, T. Hestilow, W. Haskins and Y. Huang. Review of peak detection algorithms in liquid-chromatography-mass spectrometry. *Curr. Genomics*, 10(6):388–401, 2009.

[245] J. Zhang, D. Xu, W. Gao, G. Lin and S. He. Isotope pattern vector based tandem mass spectral data calibration for improved peptide and protein identification. *Rapid Commun. Mass Spectrom.*, 23(21):3448–3456, 2009.

[246] 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, 2002.

[247] 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.

[248] R. Zubarev and M. Mann. On the proper use of mass accuracy in proteomics. *Mol. Cell. Proteomics.*, 6(3):377–381, 2007.